

TU MICRO COMMODORE
nº 9 160 ptas (IVA incluido)
SEMANAL
ELITE: PRIMERA PARTE DE UNA GRAN AVENTURA.
CODIGO MAQUINA.
DEBUTA UNA NUEVA SECCION.
RATON PARA C-64.
GANADOR DEL PRIMER CONCURSO DE PROGRAMACION.
CHISPAS: LOS TRUCOS MAS INTERESANTES PARA NUESTRO COMMODORE.

ESTA SEMANA

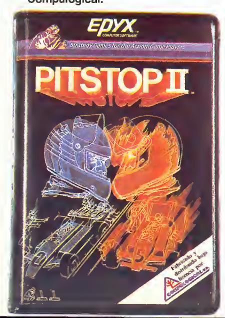
SOFTWARE UTILIZANDO LAS INTERRUPCIONES	4
EL TABLON	5
SOFTWARE CIRCULOS Y RECTAS EN ALTA RESOLUCION	11
LOS 7 MAGNIFICOS	12
CHISPAS	14
CURSO DE BASIC ESTRUCTURAS DE BUCLE	15
SOFTMODORE PITSTOP II Y SUMMER GAMES II	19
TECLA A TECLA RATON	22
FUERA ERRORES	26
CONCURSO	30

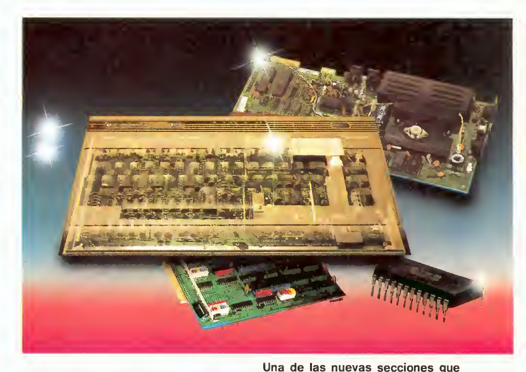
Y ADEMAS...



Solo con decir que una sola sección de A FONDO no es suficiente para el Elite ya está dicho todo. La primera parte de una emocionante aventura que continuará en nuestro próximo número.

PITSTOP II. Un programa que va algo más allá de una simple carrera de coches; otro éxito de Epyx-Compulogical.





Una de las nuevas secciones que ya anunciábamos en nuestro número pasado debuta esta semana: CODIGO MAQUINA.

UTILIZANDO LAS INTERRUPCIONES

Supón que te encuentras jugando con tu C64 a tu juego favorito, en ese momento suena el teléfono; ¿qué haces?, muy fácil, pulsas la tecla para detener el juego, contestas el teléfono, y continúas por donde lo habías dejado.

Algo muy parecido a esto, sucede en el interior de nuestro ordenador, de ello se ocupa la denominada rutina de interrupción. Cuando se ejecuta un programa, el microprocesador no sólo está ocupado en esa tarea, tiene cosas importantes que hacer como: ver si has pulsado la tecla STOP (y en general cualquier tecla), poner la pantalla al día, controlar la hora del reloj (actualizarla), ver si hay que llamar al camión de la basura (garbage collection), etc.

Esa rutina, se ejecuta 60 veces por segundo, y se encuentra en la dirección 59953 ($EA31), y el puntero que señala esta dirección está en las direcciones 788-789 ($0314-$0315). Pues se nos puede ocurrir, que cambiando este puntero para que apunte a una dirección nuestra, donde previamente habremos colocado una rutina en código máquina, esta rutina se ejecutará 60 veces por segundo, ¿no es así?. ¡¡CIERTO!!

Nosotros hemos puesto una rutina de movimiento de sprites, pero se puede colocar cualquier cosa que se desee. Para usarla, teclea el cargador BASIC que acompaña este artículo. Ahora, vamos a ver el listado Ensamblador.

Desde C000 hasta C00C, cambia el puntero de interrupción (siempre en C.M.!!), para que apunte a C00D, donde está la rutina de mover dos sprites; notemos que a esta rutina (C00D), no se le llama, simplemente se ejecuta 60 veces por segundo, teniendo la precaución de terminarla con un salto a $EA31, para que todo continúe normal.

DIR.	HEXADEC.	MNEMONICO	
C000	78	SEI	: Permite interrupciones
C001	A9 0D	LDA#$0D	: Carga A con 0D
C003	8D 14 03	STA $0314	: Carga byte bajo de puntero
C006	A9 C0	LDA#$C0	: Carga A con C0
C008	8D 15 03	STA $0315	: Carga byte alto del puntero
C00B	58	CLI	: Deshabilita interrupciones
C00C	60	RTS	: Vuelve al BASIC
C00D	A2 00	LDX#$00	: Carga 0 en X
C00F	EE 01 D0	INC $D001	: Incrementa D001
C012	CE 03 D0	DEC $D003	: Decrementa D003
C015	E8	INX	: Incrementa X
C016	E0 80	CPX#$05	: Compara X con 5
C018	D0 F5	BNE $C00F	: Si no es 5, salta a C00F
C01A	4C 31 EA	JMP $EA31	: Salta a interrupción normal




```
10 FORI=49152TO49180:READA:C=C+A:POKEI,A:NEXT                                -052-
20 IFC<>3475THENPRINT"ERROR":END                                              -060-
30 SYS49152:PRINTCHR$(147)"INTERRUPCION ACTIVADA, PARA DESACTIVAR:"           -047-
40 PRINT"RUN/STOP-RESTORE: POKE49175,VELOCIDAD"                               -090-
50 V=53248                                                                    -030-
60 POKEV+21,3:POKEV+39,0:POKEV+40,0:POKE2040,192:POKE2041,193                 -068-
70 POKEV+16,2:POKEV,24:POKEV+2,64                                             -255-
80 FORI=12288TO12414:POKEI,255:NEXT                                           -140-
100 DATA 120,169,13,141,20,3,169,192,141,21,3,88,96,162,0,238,1,208,206,3
                                                                              -052-
110 DATA 208,232,224,5,208,245,76,49,234                                      -242-

READY.                                                                        -173-
```

el tablón

TU MICRO COMMODORE
(EL TABLON)
APARTADO de CORREOS 61.294
28080 MADRID

Queremos crear un club de usuarios en Almería de usuarios de Spectrum y Commodore 64, con intercambio de programas, juegoclub, concursos, editar una revista, cursillos de informática... Interesados contactar con Miguel Angel de Rueda Ferre. Apdo. Correos 935 04080 ALMERIA; o al teléfono 340293 de 5 a 8 de la tarde.

Me gustaría comprar ordenador Commodore 64 por menos de 20.000 Ptas. y cintas de juegos por 500 Ptas.; mandadme lista. Jorge Riera Ramírez. C/ Norte 3, bajo. Port Bou, GERONA. Tfn.: 390699.

Vendo VIC-20 con dos meses de uso. Instrucciones en castellano, manual de usuario, cables, alimentación, juego cartucho Road Race por 15.000 Ptas. Llamar a Félix Gallego. Rambla de Cataluña, 5 3.° Villaseca, TARRAGONA. Tfn.: (977) 391846.

No soy «pirata», pero por razones económicas vendo programas que ya no utilizo, así como un juego de Luces Rítmicas (cuatro canales), una emisora de FM (3W) y algunas cosas más (programas para Commodore 64). Tfn.: 742 03 99 (MADRID). También deseo contactar con programadores amigos de C-64.

SOLO SANTANDER. Compro unidad a cassettes C2N, con cabezal en buen estado. Precio sobre poco más de 3.000 Ptas. Interesados escribid a Eduardo Casino. C/ Joaquín Costa, 23. 39005 SANTANDER.

Vendo VIC-20 en perfecto estado, un año por 15.000 Ptas. Vendo 30 revistas de Commodore más 40 programas en cinta. Llamar a Pedro. Tfno.: 29 13 82. C/ Renedo, 14 1.° 47005 VALLADOLID.

Quisiera contactar con club de usuarios de Commodore 64 de Valladolid, o provincias de alrededor, podría proporcionar algunas cosas. Preguntar por Santi. Tfn. 255281 de 2 a 3 de la tarde.

Se vende para VIC-20 dos cartuchos ajedrez SARGON II, RADAR RATE RACE, Curso de BASIC I y II, guía del usuario y una cinta con juegos, todo por 6.000 Ptas. Luis Griell Gil. C/ Mayor de Gracia, 118 4.°. Tfno. 218 17 18 de 3 a 8 de la tarde.

Vendo Commodore 16 como nuevo (ENERO 85), casete, cables, manuales. El mejor BASIC del mercado (do, loop, until, while, circle, else, etc) dos cintas originales, una de ellas la más vendida en UK (BMX RACERS), lote de revistas de Commodore, monitor de código máquina incluido en ROM. Todo por 38.000 Ptas. Narciso Mesa Bravo. Ronda del Salvador, 85. Plasencia, CACERES. Tfno. 411667.

CBM 64. Intercambio programas. Poseo unos 700. Mandar lista a Javier Hernández Sole. C/ El Roser, 34. Mora la Nova, TARRAGONA. Tfn.: 401084.

Intercambio programas en cinta para C-64. Interesados enviar lista. Jordi Aymerich. C/ Trebol, 7. Sarriá de Ter, GIRONA. Tfn.: (972) 209755.

Vendo Commodore 64, unidad lectora de casetes, manual de instrucciones, dos tomos de un curso de BASIC, la guía de referencia, muchas revistas, documentos y garantía, más de 90 programas en turbo, juegos aplicaciones, lenguajes... todo a precio interesante. Llamar preguntando por Fernando, al teléfono de BARCELONA (93) 254 92 72.

Cambio o compro las siguientes revistas Commodore World (N.os 1 al 5 y 7) o listados de programas interesantes, preferentemente musicales. También vendo las siguientes revistas en cinta C-64 Tape Computing (n.° 1) y Commodore Computing Club (n.° 1). Precio a convenir. También intercambio programas en cinta (tengo pocos). Dirigirse a Antonio Saa Quintas. El almendro, 120. Aguadulce, ALMERIA. Tfn. (951) 340293. Preguntar por Toni de 16 a 20 h.

Vendo impresora Commodore (SEIKOSHA 250X) seminueva por 25.000 ptas. Interesados llamad a horas de comidas al 445 83 35, prefijo 91 para llamadas de fuera de Madrid. Preguntar por José Luis.

Compro ordenadores estropeados. Desde 3.000 hasta 6.000 pesetas. Da lo mismo la marca. Pedro David Casardo Santiago. C/ Padre Claret 50-62 4.° 2.ª escalera derecha BARCELONA. Teléfono 258 77 40. Después de las 9 de la noche.

Club (C) LDA. Intercambios a todo nivel para 64, 128 y Amiga. Extensa biblioteca de soft original y posesión de hard. Para contactar: Andrey Ibáñez i Perales. C/ Passeig de Ronda 76, 3-(25006), Lleida (para clubs y particulares).

Vendo VIC-20 con fuente de alimentación, unidad de ciclo y pantalla, con libros de programación, todo muy nuevo, sin usar (2 semanas); a buen precio. Harald Schillarg. C/ Cuatro 12, Castelldefels. (Barcelona). Tfno. 93-665 24 33.

¿Quieres participar en un club de estudiantes de B.U.P. que crea sus propios programas para ayudarse en los estudios? Escribidnos a C.E.B. CO-64. C/ Virgen de los Milagros 96, Puerto de Santamaría. (Cádiz).

Tengo un VIC-20. Quisiera establecer contactos con otras personas que también lo tengan, para intercambiar programas. Pedro Simona. C/ Portal de Cerdana. Seu de Urgen (Lérida).

Intercambio programas en cinta para Commodore 64. Mandad lista, prometo contestar. Cristóbal Dávila Frances. C/ Odena, 36, 3.°. Igualada (Barcelona).

Atención commodorianos, desearía intercambiar programas para C64, juegos y utilidades. Contesto a todas las cartas. Albert Casellas i Tusal. C/ Carme 9. Igualada (Barcelona).

Vendo VIC-20, fuente de alimentación, dos cartuchos, radar ratace, Sargon II Chess, dos cintas de iniciación, tres libros de programación, varias cintas de juegos, todo por 20.000 Ptas. Preguntar por Manolo. C/ Tenerife 27, Entresuelo 1. Tfno.: 219 72 07. Barcelona 08024.

Si quereis intercambiar programas, experiencias, trucos, proyectos e iniciativas para C-64, escribir al Club Intersoft. Apartado de Correos 27016-08020 Barcelona; o a Manuel Arroyo, C/ Maresma 266, Barcelona. Entra en el primer Club de intercambio de iniciativas y programas para el C-64.

Atención commodorianos, desearía intercambiar programas para C-64, juegos y utilidades. Contesto a todas las cartas. Albert Casellas i Tusal, calle Carme, 9, Igualada (Barcelona).

Tengo un VIC-20. Quisiera establecer contactos con otras personas que también lo tengan, para intercambiar programas. Pedro Simona, Calle Portal de Cerdana. Seu de Urgel (Lérida).

Intercambio programas en cinta para Commodore 64. Mandad lista, prometo contestar. Cristóbal Dávila Francés. Calle Ordena, 36-3. Igualada (Barcelona).

Vendo VIC-20, fuente de alimentación, dos cartuchos, radar ratace, Sargón II gos, todo por 20.000 pesetas. Tres libros de programación, varias cintas de juegos, todo por 2.000 pesetas. Preguntar por Manolo, calle Tenerife, 27. Entresuelo 1, teléfono: 219 72 07, Barcelona 08024.

Vendo impresora para Commodore (SEISKOSHA 250 X), seminueva, por 25.000 pesetas. Interesados llamad a horas de comidas al 445 83 35, prefijo 91 para llamadas de fuera de Madrid. Preguntar por José Luis.

Si queréis intercambiar programas, experiencias, trucos, proyectos e iniciativas para C-64, escribir al Club Intersoft. Apartado de Correos 27016-08020 Barcelona; o a Manuel Arroyo, calle Maresma, 266, Barcelona. Entra en el primer Club de intercambio de iniciativas y programas para el C-64.

ELITE (I)

Mientras los filósofos, científicos y sabios de todas clases conjeturan sobre el Universo, dos programadores: IAN BELL y DAVID BRAVEN, han conseguido introducirlo en la memoria del COMMODORE 64.

ELITE es un espectacular programa basado en la novela de ciencia-ficción «The Dark Wheel» («La rueda oscura»), y más concretamente en las andanzas de su personaje principal, Alex Ryder.

Alex Ryder realizó su primer vuelo pegado a los pantalones de su padre. Cogido de la mano de éste, observaba boquiabierto el espacio a través de las ventanillas de la nave espacial. También observaba boquiabierto a la azafata de a bordo, mujer de notable destreza, con la cual sostenía su padre una relación poco lícita. Poco después de despegar, una nave se aproximó hacia ellos rauda y velozmente, disparando a discreción. Aunque Discrección no resultó herido, si lo fue el piloto Jason Ryder. Alex, por el contrario, se deslizó dentro de una cápsula de escape y consiguió sobrevivir a la catástrofe.

Entre los restos de la nave de Jason se halló la caja negra grabada con las últimas palabras del piloto. La causa de la tragedia fue la imposibilidad de escapar del ataque del ofensor, debido a la acción del chicle que Alex tirara descuidadamente, el cual se introdujo en el cambio de marchas de la aeronave.

A bordo de una nave hospital, Alex recuperó el sentido y padeció «resaca espacial», es decir, los síntomas producidos al navegar durante un largo período de tiempo. Varias tazas de café y unas horas más tarde, se propuso seguir el camino de su padre y completar su misión, llegando a ser un miembro de la «ELITE».

Alex se hizo entonces con una aeronave y se lanzó al espacio a probar fortuna en el comercio interestelar. Aún afectado por la muerte de su padre, se negó a comer chicle hasta el ansiado momento en el que fuera nombrado «ELITE» (¿lo conseguirá?).

NUESTRA RAZON DE VIDA: ALCANZAR LA ELITE. ALGO NADA SENCILLO DE CONSEGUIR

Verdaderamente no es difícil captar el objetivo del juego, claramente expresado en la novela, el manual de instrucciones y la pantalla de presentación visualizada en el transcurso de la carga del programa. Pues sí, un diez para aquellos que lo hayan adivinado: la misión de Alex, encarnado por el usuario, es la consecución del rango de «ELITE». Esta tarea es ardua y, desde luego, exige el empleo de mucho tiempo de pilotaje a los mandos de nuestra aeronave.

El rango se obtiene de acuerdo con las habilidades del piloto para el comercio. El programa evalúa positiva o negativamente las operaciones económicas desarrolladas por el jugador y cuando se reúnen suficientes puntos positivos se consigue un aumento de grado; en caso contrario, desciende el rango del piloto.

En orden de importancia existen nueve niveles: Harmless, Mostly Harmless, Poor, Average, Above Average, Competent, Dangerous, Deadly y ELITE. En un principio, nuestro rango será el de Harmless, siendo necesario atravesar ocho grados para conseguir el ansiado objetivo.

MANUAL DE USO DEL PERFECTO PILOTO DE LA COBRA MK III

No se puede hablar de la buena estrella del programa, sino de las 2.000 buenas estrellas. Resultaría muy cansado recorrer todas las estrellas del Universo a pie, con unas bolsas de plástico en las manos donde guardar la mercancia. Para evitarnos estas molestias, disponemos de una astronave intergaláctica, una auténtica máquina de precisión: la Cobra MK III.

El control de la Cobra depende del uso del volante y del acelerador. El volante desplaza la nave hacia cuatro direcciones posibles: arriba (tecla S o joystick), abajo (tecla X o joystick), izquierda (, < o joystick), y derecha (. > o joystick). La velocidad es regulada mediante la tecla ?/ para acelerar,


POLO 86
MOVIMIENTO
DE LA COBRA MKII
Movimiento lateral. No hay desplazamiento efectivo
Movimiento vertical.
La nave se desplaza

y la barra espaciadora para frenar. Con el fin de orientar la nave hacia un punto, basta situar a este por encima de nosotros con giros a izquierda o derecha, y luego centrarlo en la pantalla bajando con la nave.

Se trata de una nave mercantil muy popular, pues combina una gran capacidad de combate y mucho espacio para el cargamento. La máxima carga transportable en condiciones normales es de 20 toneladas, aunque puede ser ampliada. Es un transporte veloz, muy maniobrable y capaz de sostener una gran cantidad de equipos suplementarios (EQUIP SHIP).

Dentro de la cabina del piloto encontramos sensores y sofisticados sistemas técnicos. Cada uno de estos sensores tiene una función diferente, veamos cuales son:

1. Los escudos que defienden la nave toman su energía de estos dos blancos. Si un valor es nulo, los escudos no se regeneran. Reduciendo la velocidad de la nave, los escudos y los bancos de energía se recargan con mayor rapidez.

2. Contador de fuel. Muestra la cantidad de fuel disponible. Carece de utilidad, pero hace bonito.

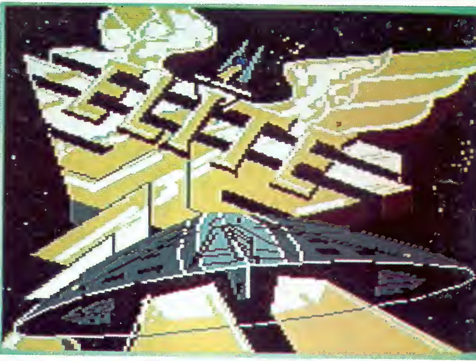
3. Temperatura de la cabina. Cuando la nave se sitúa próxima a un sol, la temperatura de la cabina aumenta. Una temperatura muy elevada, aparte de contribuir al bronceado del piloto, destruye la carga y los sistemas de navegación.

4. Temperatura del laser. Si los laser son disparados muy rápidamente, se recalientan y disminuye su cadencia de disparo.

5. El altímetro muestra la distancia de la nave respecto de la superficie del planeta. Si desciende de mucho nos estrellaremos contra éste.

6. Representa el número de misiles restante y, por oposición, los que ya han sido disparados. Señala la existencia de un misil cargado con el cambio de color de un cuadrado.

7. Radar tridimensional donde quedan reflejados todos los móviles del espacio próximos a nosotros. Nuestra nave está situada en el centro del radar, las demás naves están representadas por barras con un punto en su extremo. Si la barra queda por encima del plano del radar, el móvil se encuentra en una posición superior. Análogamente, para indicar una posición inferior del móvil, la barra está representada por debajo del plano del radar. Una barra de color oscuro señala la posición de



EN VUELO:	
GIRO HACIA LA DERECHA	> o joystick
GIRO HACIA LA IZQUIERDA	< o joystick
ARRIBA	S o joystick
ABAJO	X o joystick
ACELERADOR	(barra espaciadora)
FRENO	/?
VISTA FRONTAL	F1
VISTA TRASERA	F3
VISTA IZQUIERDA	F5
VISTA DERECHA	F7
DISPARO DE LASER	A o joystick
CARGA MISIL	T
DESARMAR MISIL	U
DISPARAR MISIL	M
ACTIVAR ECM	E
BOMBA DE ENERGIA	C=

CAPSULA DE ESCAPE	←
COMPUTADOR DE ATERRIZAJE CONECTADO	C
IDEM DESCONECTADO	P
NAVEGACION:	
HIPERESPACIO	H
SALTO INTERGALACTICO	CTRL + H
DISTANCIA	D
CURSOR AL ORIGEN	Q
MOVIMIENTO DEL CURSOR	←CSR→ CSR
CARTA GALACTICA	4
CARTA LOCAL	5
COMERCIO:	
DESPEGAR	F1
COMPRAR	1

VENCER	2
EQUIPOS	3
DATOS DEL PLANETA	6
PRECIOS DEL PLANETA	7
STATUS	8
INVENTARIO	9
AYUDA AL JUGADOR:	INST DEL
PARA EL JUEGO	CLR HOME
CONTINUA EL JUEG	
GRABAR / EXTRAER	@
MUSICA DE ATERRIZAJE SI O NO	M
DESCONECTAR EFECTOS DE SONIDO	Q
CONECTAR EFECTOS DE SONIDO	S
LINEAS DEL PLANETA	P

un asteroide y una de color blanco, la de otra nave.

8. El girocompás ayudará en la orientación hacia el planeta o la base espacial centrando el cursor en el medio. ¡Precaución!, la trayectoria sólo es correcta si el cursor aparece como un punto grueso, no como un guión, en cuyo caso el objetivo queda detrás nuestro.

9. El contador de velocidad.

10. Digitalizadores de giro. Al igual que el contador de fuel, carecen de utilidad práctica, pero también decoran.

11. Escudos de protección. Si estas barras desaparecen completamente, entonces ¡PUM!, sin más comentarios.

El radar es el instrumento de vuelo más útil; gracias a él localizamos todo artefacto móvil y podemos visualizarlo en la pantalla. Las naves situadas en el mismo plano horizontal que la Cobra pueden ser observadas con el uso de las cuatro vistas o panorámicas de la nave: Frontal (**F1**), Trasera (**F3**), Izquierda (**F5**) y Derecha (**F7**).

Hemos hablado acerca del viaje entre planetas, pero en realidad, la nave nunca pisa el suelo de un planeta; no porque haya sido encerado hace poco, sino porque aterriza en una estación espacial situada cerca de éste. Las estaciones tienen dos diseños distintos: hexagonal (estación CORIOLIS) y dodecaédrica (estación DODO). Estos aparcamientos espaciales son lugares neutrales donde se compran y venden, salvo excepciones, cualquiera de los 17 productos del mercado intergaláctico. También son comprados en estas zonas los equipos de vuelo (EQUIP SHIP) y los desperfectos sufridos por la nave son reparados, restableciéndose la fuerza de los escudos completamente.

Todos los logros de un piloto se muestran en su STATUS, accesible mediante la tecla **8**. Se obtiene en cualquier fase de la partida y revela los siguientes datos:

- Nombre del comandante.
- Present System. El nombre del último planeta visitado por la espacionave.
- Hyperspace System. El planeta sobre el cual hayamos situado la cruz (el cursor de los mapas galácticos, como veremos más adelante). Generalmente coincide con el destino de nuestro viaje.
- Condition. Permite vislumbrar la situación de la nave, si se encuentra posada en una base (Docked) o no; en este último caso admite tres posibles valores: Green, Yellow o Red (verde, amarilla o roja), a modo de semáforo. Cuando no existen atacantes en las proximidades tendremos condición verde, amarilla en caso de existir enemigos que no hayan atacado todavía, y rojo cuando nos encontremos en plena lucha.
- Fuel. Señala la distancia en años luz que es posible recorrer con el combustible restante.
- Cash. Es la pasta, el money, los cuartos, la guita, los créditos disponibles.
- Legal Status. Indica la situación del piloto respecto de la policía. Toma tres valores: Clean, Offender y Fugitive. En función del estado legal, un piloto será atacado (Offender, Fugitive) o no (Clean) por los vipers policiales. La variación del Legal Status se produce al atacar fuerzas policiales o comerciar con algunos productos ilegales del mercado (Esclavos, Narcóticos y Armas de Fuego).
- Rating. Desde Harmless hasta Elite este marcador refleja cualquiera de los nueve posibles grados del piloto mencionados con anterioridad.
- Equipment. El último marcador del STATUS señala el equipo de navegación (EQUIP SHIP) del cual disponemos.

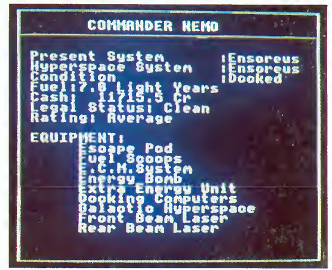
En cualquier momento de la partida, al pulsar la tecla '9' (INVENTORY), el cargamento transportado por nuestra nave es mostrado en la pantalla, además del fuel y del dinero disponibles.

La Cobra MK III es capaz de realizar travesías espaciales entre planetas situados a una distancia máxima de siete años luz, pero gracias al sistema de salto intergaláctico aparece la posibilidad de cambiar de galaxia. Los saltos, efectuados entre las ocho galaxias

LOS CUATRO ULTIMOS COMANDOS SON ACTIVABLES PREVIA DETENCION DEL JUEGO CON LA TECLA INST/DEL

CABINA DEL PILOTO

1. BANCOS DE ENERGIA.
2. CONTADOR DEL FUEL.
3. TEMPERATURA DE LA CABINA.
4. TEMPERATURA DEL LASER.
5. ALTIMETRO.
6. MISILES.
7. RADAR.
8. GIROCOMPAS.
9. VELOCIDAD
10. DIGITALIZADORES DE GIRO.
11. ESCUDOS.



conocidas, se producen de forma ascendente y descendente, es decir, de la galaxia 1 saltamos a la 2, de esta a la 3, y así hasta llegar a la octava, desde la cual, con sucesivos saltos, atravesaremos en sentido inverso las galaxias recorridas anteriormente.

Cada galaxia tiene asignado un mapa donde están representados los 250 mundos de dicho sistema y dos cruces de distinto tamaño. La cruz más grande muestra la posición de la Cobra, y la más pequeña (el cursor) está situada sobre cualquier punto elegido por el piloto, ya sea como destino o para informarse sobre los datos del mismo. El control de una carta galáctica, obtenible con la tecla **4**, se materializa con el uso de un conjunto de comandos asociados a esta:

• Las teclas de cursores o el joystick desplazan el cursor en cualquier dirección. Presionando el botón de disparo se aceleran los movimientos de la cruz.

• Ambas cruces coinciden en el mismo punto (la posición actual de la nave) al pulsar la tecla **0**.

• Para ayudarnos a interpretar la carta (galáctica, no de un restaurante) contamos con una carta de corto alcance (tecla '5') que muestra un radio aproximado de unos 10 años luz. En el centro de esta carta está situada la cruz de mayor tamaño, rodeada por una circunferencia tanto mayor cuanto mayor es la cantidad de fuel que disponemos; éste es el límite de nuestro radio de acción.

• Aparte, la distancia entre la cruz grande y la pequeña se calcula al presionar la **D**.

• La tecla **6** representa en la pantalla los datos del planeta señalado por el cursor. Estos datos son de dos tipos: importantes y anecdóticos. Los primeros (distancia, economía, gobierno y nivel técnico) serán tenidos muy en cuenta por el jugador. Los demás, son datos suplementarios sin ninguna utilidad.

El comercio entre planetas es la parte fundamental del juego y la única que permite aumentar progresivamente el rango del piloto hasta llegar a la «ELITE». Son utilizadas tres instrucciones para las transacciones comerciales, accesible, como no, mediante la pulsación de sus respectivas teclas, para variar. Estos comandos son:

• Tecla **7**, Market Prices o Precios de Mercado. Es una relación encolumnada de los 17 productos con los cuales pueden realizarse operaciones económicas, sus precios y las existencias del planeta donde estemos posados.

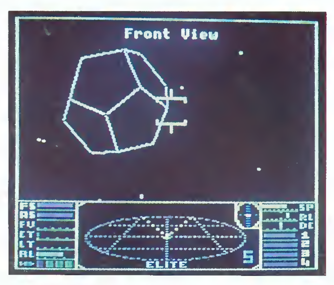
• Tecla **1**, Buy Cargo o Compra de Carga. El cargamento se adquiere, previo pago de su costo, con esta función.

• Tecla **2**, Sell Cargo o Venta de Carga. Si el precio de venta es superior al precio de compra venderemos nuestro cargamento o parte de éste.

Cansados ya de reposar en la estación espacial, seleccionamos un planeta alcanzable por la nave y nos lanzamos al vacío (no tan vacío, más bien ocupado); pulsando **F1** saldremos de la estación espacial. El primer paso es saltar al hiperespacio normal (**H**) o al hiperespacio intergaláctico (**CTRL + H**).

Al salir del salto, el planeta siempre estará situado hacia la derecha y por encima de la nave. Situando el planeta en el centro del tablero de mandos, repetidas pulsaciones de la tecla **J** nos llevarán más rápidamente hacia nuestro objetivo. En una posición próxima al planeta aparecerá una S en el tablero de mandos, momento en el que debemos orientarnos hacia la estación espacial. Finalizaremos el vuelo introduciendo la nave en dicha estación con mucha delicadeza o con el Docking Computer u Ordenador de Aterrizaje, el cual realizará esta tarea por nosotros mientras deleita nuestros oídos con los compases del Vals del Danubio Azul por si nos apetece bailar con nuestro C-64.

La próxima semana continuaremos con esta apasionante aventura.





RECTAS Y CIRCULOS EN ALTA RESOLUCION

Siempre es interesante disponer de las sentencias LINE, DRAW, o cualquier otra que trace rectas en alta resolución, sobre la pantalla de nuestro ordenador; o de la sentencia CIRCLE, que hace ló propio con los círculos.

Desgraciadamente, en el BASIC 2.0 del C64, no disponemos de tales sentencias, y para emularlas, tendremos que recurrir a POKEs y PEEKs, lo cual resultará algo lento, aunque el resultado final es el mismo.

El programa que acompaña a este artículo, dibuja una recta o un círculo según nuestra elección. Consta de cuatro partes bien diferenciadas:

1: Líneas 10-60, se encargan de elegir el color de la tinta (t), el color del papel (f) y de la elección (recta o círculo); así como de la limpieza de la pantalla, entrada en alta resolución y asignación de colores.

2: Líneas 5050-5110, se encargan de trazar el círculo. Las coordenadas (xx,yy), son las del centro del círculo; el radio se puede cambiar también alterando la variable (ra).

3: Líneas 6050-6160, son las encargadas de trazar la recta desde el punto (cc,rr) hasta el (xx,yy); cambiando estos valores (línea 6050), obtendremos rectas diferentes.

4: Líneas 10000 hasta el final, rutina de trazado. Esta rutina se encarga de trazar los puntos en alta resolución, para formar la recta o el círculo elegido.

Empleando estas rutinas en nuestros propios programas, podremos dibujar rectas y curvas.

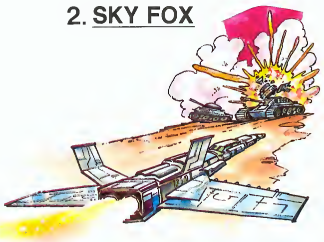
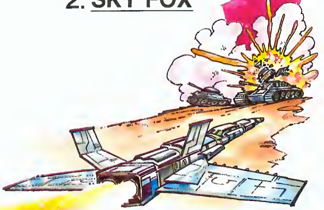



```
10 INPUT"{CLR}COLOR DE LA TINTA";T                                      -226-
20 INPUT"{ABJ}COLOR DEL PAPEL";F                                        -238-
30 INPUT"{ABJ}{RON}R{ROF}ECTA O {RON}C{ROF}IRCULO";A$                   -095-
32 POKE53265,PEEK(53265)OR32                                            -102-
34 POKE53272,PEEK(53272)OR8                                             -055-
36 FORI=8192TO16191:POKEI,0:NEXT                                        -246-
38 FORI=1024TO2023:POKEI,F+16*T:NEXTI                                   -031-
40 IFA$="R"THEN6050                                                     -069-
50 IFA$="C"THEN5050                                                     -054-
60 GOTO30                                                               -034-
5050 PP=2* :XX=160:YY=100:RA=55                                         -014-
5060 FORJ=0TOPPSTEPPP/360                                               -112-
5070 C=XX+RA*COS(J)                                                     -132-
5080 R=YY+RA*SIN(J)                                                     -155-
5090 GOSUB10000                                                         -095-
5100 NEXTJ                                                              -111-
5110 GOTO5110                                                           -221-
6050 CC=10:RR=20:XX=160:YY=100:IFABS(XX) ABS(YY)THEN6110                -176-
6060 FORJ=CCTOCC+XXSTEPSGN(XX)                                          -009-
6070 C=J:IFXX=0THENR=RR:GOTO6090                                        -065-
6080 R=YY/XX*(J-CC)+RR                                                  -085-
6090 GOSUB10000                                                         -096-
6100 NEXTJ:GOTO6160                                                     -176-
6110 FORJ=RRTORR+YYSTEPSGN(YY):R=J                                      -088-
6120 IFYY=0THENC=CC:GOTO6140                                            -010-
6130 C=XX/YY*(J-RR)+CC                                                  -066-
6140 GOSUB10000                                                         -092-
6150 NEXTJ                                                              -117-
6160 GOTO6160                                                           -243-
10000 BY=8192+320*INT(R/8)+8*INT(C/8)+(R AND 7)                         -039-
10010 BI=7-(C AND 7)                                                    -028-
10020 POKEBY,PEEK(BY)OR(2 BI)                                           -039-
10030 RETURN                                                            -244-
```



Colaboremos todos en la confección de estos 7 MAGNIFICOS de TU MICRO COMMODORE. Envíanos el nombre de tu programa favorito, dejando muy claro tu nombre y dirección. Todas las semanas sortearemos una suscripción por un año a nuestra revista entre las respuestas recibidas. Anímate y escríbenos a: TU MICRO COMMODORE. (7 MAGNIFICOS). APARTADO DE CORREOS 61.294. 28080 MADRID.

2. SKY FOX

3. KARATEKA



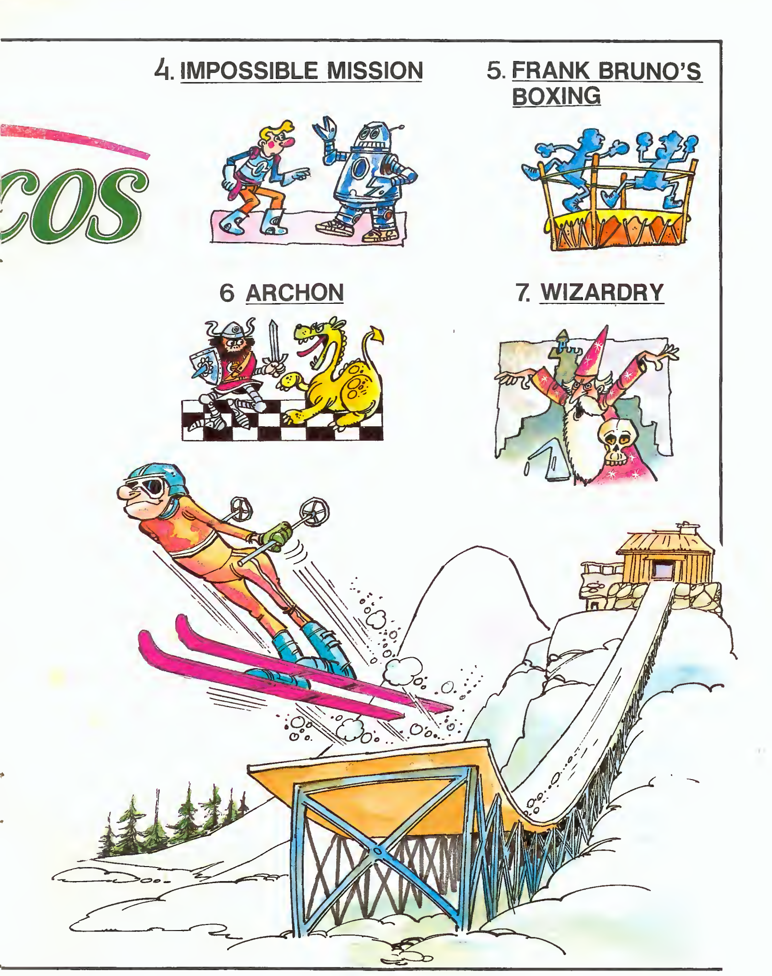




ESTA SEMANA

WINTER GAMES

El intenso frío reinante en estos salvajes parajes sitos en Calgary, Estado de Alberta, Canadá, no impide que nuestro ánimo se encuentre en su punto más álgido, y que no sintamos perfectamente preparados para competir en las más duras disciplinas olímpicas de estos Juegos de Invierno de 1988. En WINTER GAMES tendremos como oponentes a los mejores atletas representantes de otras muchas naciones, deseosos como nosotros de proporcionar la gloria de los laureles a sus respectivos países y ver inscritos sus nombres en la historia de las Olimpiadas.


4. IMPOSSIBLE MISSION
5. FRANK BRUNO'S BOXING
COS
6 ARCHON
7. WIZARDRY

chispas

CHISPA

A los usuarios de una unidad de disco, nos resultará familiar la carga de un programa cuyo nombre no recordamos exactamente, de esta manera: LOAD "ZO*",8. Este comando cargará el primer programa del directorio, cuyo nombre comience con las letras "ZO", como "ZOOM" o "ZOO" o "ZOO LOGICO"; pues bien, lo que quizás no sea tan conocido, es que los usuarios del datassette, pueden conseguir el mismo efecto con LOAD "ZO",1. Esto cargará el primer programa de la cinta, que comience por las letras "ZO".

CHISPA

Hay veces en que necesitamos ver el contenido de una cierta zona de memoria, o el valor de una variable de matriz bastante larga, o cualquier otra cosa de bastante longitud; pero debemos hacerlo en modo directo (si pulsamos 'RETURN' para introducir una línea de programa las variables se ponen a cero), entonces nos encontramos con el problema de detener la impresión en un bucle de este tipo: FOR I=1 TO 100:PRINT A$(I): NEXT. Una solución podría ser detener el listado hasta que se pulse una tecla, pero GET no es ejecutable en modo directo. Pues bien, he aquí la solución: FOR I=1 TO 100:POKE198,0WAIT 198,1:PRINTA$(I):NEXT. Esto detendrá cada ejecución del ciclo hasta que se pulse una tecla.

CHISPA

El buffer de la grabadora comienza en la posición de memoria 828 y termina en la 1019. Cuando cargamos un programa, tenemos información interesante en este buffer que podemos conocer. Para saber donde comienza el programa y donde termina (muy útil si es un programa en código máquina), podemos usar: FOR I=829TO832:PRINTO,PEEK (I):NEXT, donde los valores de 829 y los punteros de fin (algunos programas comerciales, tiene estos punteros en 828-831). Y en las posiciones 833-849, tenemos el nombre del programa, para verlo usamos: FORI=833TO 849:PRINTCHR$(PEEK(I));: NEXT.

CHISPA

Algunos equipos incorporan un tipo de sentencia que sirve para posicionar el cursor en unas determinadas coordenadas de la pantalla, como puede ser PRINT AT, LOCATE, u otras, que cumplen función similar. En el C64 no tenemos ninguna sentencia que haga esto, pero disponemos de ciertas rutinas del sistema operativo, que pueden utilizarse para ello. Para aprovecharlas, debemos ejecutar desde BASIC la siguiente subrutina: 100POKE211,X:POKE 214,Y:SYS58732 y la podremos usar así: 10X=8: Y=15:GOSUB100:PRINT" MENSAJE". Donde X es el número de la columna (margen izquierdo = 0), e Y es la fila (HOME = 0), y "MENSAJE" es el texto a imprimir en el lugar deseado.

CHISPA

Para eliminar el carácter "?" que acompaña a cada INPUT, y conseguir que las pantallas queden más vistosas y presentables, deberemos hacer lo siguiente: POKE19,1:INPUT......:POKE19 ,O:PRINT. O sustituir el INPUT normal por: OPEN1,0: INPUT 1......:PRINT:CLOSE1. Esto último abre el teclado como canal de entrada, lo cual, aunque redundante, consigue el efecto deseado.

CHISPA

Existen algunos códigos ASCII en el C-64 que no disponen de definición en el manual, uno de éstos es el 131, y es un carácter verdaderamente curioso, puesto que ofrece el efecto conseguido por las teclas SHIFT+RUN/STOP. Para probarlo necesitamos situarlo directamente en el buffer del teclado (no sirve PRINT CHR$(131)), de la siguiente forma: POKE631,147: POKE632,131:POKE633,13:POKE198,3 así conseguimos hacer un LOAD desde casete con AUTORUN.

CHISPA

El carácter (:) en el C64, no producirá error alguno en ningún lugar de una línea BASIC, esto se puede aprovechar para «indentar» programas, que resultarán más claros, sobre todo en la fase de desarrollo. Por ejemplo:

```
10 FORI=1TO10
20 :::PRINTI
30 NEXT I
```

Aquí resaltamos un bucle FOR...NEXT, pero se puede usar para subrutinas, bucles artificiales, parte principal... etc.

CHISPA

Si por accidente, o por medio de algún programa que lo incorpore, se ejecuta el terrorífico NEW, no te preocupes, ¡tiene solución!, antes de hacer otra cosa, ejecuta en modo directo:

POKE2050,1:SYS42291:POKE45,PEEK(34):POKE46, PEEK(35):CLR.

CHISPA

Si quieres que durante un listado no aparezcan los números de línea, teclea: **POKE22,35 y POKE22,25,** para que vuelvan a aparecer.

CHISPA

Para cambiar el color de los caracteres, no hay porque hacer un **PRINT** y pulsar el **CTRL** o **COMMODORE** del color entre comillas, también se puede conseguir el mismo efecto efectuando **POKE** a 646 con el número del color deseado.

CURSO DE BASIC

ESTRUCTURAS DE BUCLE

La ejecución de un programa implica el desarrollo de tres apartados: entrada de datos, proceso de los mismos y salida de resultados. Pues bien, dentro de esta estructura de conjunto, se plantea la necesidad de codificar de forma abreviada bloques de instrucciones cuya ejecución ha de repetirse una y otra vez, para diferentes valores de determinadas variables. Estas estructuras de programación se denominan «bucles».

Veremos más claro todo esto con un ejemplo. Supongamos que deseamos imprimir en pantalla los veinte primeros enteros positivos y sus respectivos cuadrados. Es evidente que la solución más inmediata al problema sería:

```
10 PRINT 1,1 ↑ 2
20 PRINT 2,2 ↑ 2
...
190 PRINT 19,19 ↑ 2
200 PRINT 20,20 ↑ 2
```

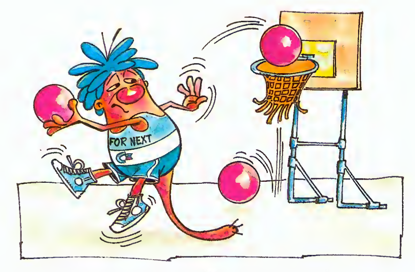
Sin embargo, esta solución no es la ideal, pues existe en BASIC una forma de codificación muy útil cuando tratamos de repetir, determinado número de veces, una instrucción concreta o bloque de ellas, para sucesivos valores de una variable. A esta estructura de codificación se le da el nombre de bucle **FOR-NEXT** y su formato general es:

```
FOR variable=comienzo TO final STEP incremento
...
NEX variable
```

Comprenderemos mejor el funcionamiento de este sistema apli-



La ejecución de un programa implica el desarrollo de tres apartados entrada de datos, proceso de los mismos y salida de resultados.



El bucle **FOR-NEXT** *es una forma de codificación en BASIC muy útil cuando tratamos de repetir una instrucción concreta*

cándolo al ejemplo anteriormente propuesto, cuya codificación podría ser:

```
10 FOR I=1 TO 20 STEP 1
20 PRINT I, I   2
30 NEXT I
```

Cuando el programa llega por primera vez a la línea 10 se asigna el valor 1 a la variable **I**. A continuación, se ejecutan los cálculos indicados en la instrucción 20. Por último, se alcanza el **NEXT** (siguiente) de la línea 30, donde se incrementa automáticamente al valor de **I** en lo indicado por la palabra clave **STEP.** En este punto, se efectúa un control para comprobar si el nuevo valor de **I** rebasa o no el límite superior indicado en la línea 10 (20) y, al resultar este inferior, el programa bifurca de nuevo a la línea 10, para efectuar una segunda pasada.

Los ciclos para valores de **I** entre 2 y 19 resultarán idénticos, produciéndose por cada pasada la impresión en la pantalla de un nuevo valor y su cuadrado. Sin embargo, al alcanzar la instrucción 30 después de la impresión de 20 y su cuadrado, la sentencia **NEXT** aplicará a la variable **I** el valor 21, excediendo por tanto el máximo indicado en la línea 10 (20). Por ello, el programa abandonará la estructura de bucle, continuando la ejecución a partir de la próxima instrucción, o finalizando si la instrucción 30 fuese la última del programa.

Generalizando, podemos decir que la palabra clave **FOR** (en inglés, «para») indica el comienzo de una estructura de bucle, de manera que la variable indicada a continuación oscilará entre determinados márgenes. El primero de ellos, se determina con el argumento que sigue al símbolo de igualdad, el cual puede ser cualquier variable o constante numérica, y el segundo viene dado por la expresión que sigue a la palabra clave **TO** (hasta).

Hasta ahora, tenemos información sobre la variable sujeto del bucle, así como sobre sus límites inferior y superior. Sin embargo, es necesario una información complementaria, que contenga el incremento, positivo o negativo, que es preciso aplicar por cada nuevo ciclo. Este incremento viene dado por el argumento que sigue a la palabra clave **STEP** (en inglés «paso»), pudiendo ser cualquier variable o constante numérica. No obstante, y dado que el empleo del incremento en la unidad es uno de los casos más generales de aplicación de los bucles, es posible omitir el calificativo **STEP** cuando el incremento deseado para la variable sea la unidad. Por ello, el programa anterior puede escribirse también como sigue:



En un bucle es necesaria una información complementaria que contenga el incremento positivo o negativo que es preciso aplicar por cada nuevo precio.

```
10 FOR I=1 TO 20
20 PRINT I,,I   2
30 NEXT I
```

NEXT (en inglés, «siguiente») tiene la misión de aplicar el incremento a la variable y comprobar el desbordamiento del límite

NEXT *tiene la misión de aplicar el incremento a la variable y comprobar el desbordamiento del límite superior.*

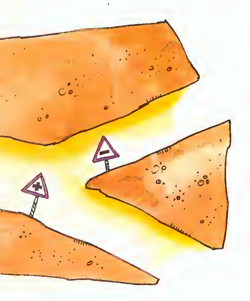
superior, llevando como argumento el nombre de la variable a la que debe afectar. Sin embargo, el BASIC del COMMODORE 64 permite omitir el nombre de esta, considerándose cada sentencia **NEXT** aplicable a la última variable de **FOR** utilizada. Sabido esto podemos escribir:

```
10 FOR I=1 TO 20
20 PRINT I,,I   2
30 NEXT
```

La doble coma (,,) de la instrucción 20 indica una tabulación al centro de la pantalla. Como ya hemos comentado, cada signo de



La doble coma (,,) asociada a una sentencia **PRINT** *produce una tabulación al centro de la pantalla*



puntuación coma (,) asociado a una sentencia **PRINT** produce un salto automático a la siguiente posición de tabulación (uno de los comienzos de cada cuarto de línea). Por ello, si especificamos dos comas (,,) obtendremos la impresión a partir del centro de la pantalla (columna 20) hacia la derecha.

APLICACIONES DE STEP

Aunque en la mayoría de los casos el **STEP** aplicado a los bucles **FOR-NEXT** es la unidad, este puede adoptar cualquier valor: positivo, negativo o decimal. El siguiente programa nos muestra una «cuenta atrás» por la aplicación de un **STEP** negativo.

```
10 FOR I=20 TO 1 STEP -1
20 PRINT I
30 NEXT
```

Veamos ahora la obtención de los primeros números pares por aplicación de un **STEP 2:**

```
10 FOR I=2 TO 40 STEP 2
20 PRINT I
30 NEXT
```

Vamos ahora a mejorar un poco la presentación anterior de forma que podamos mantener en la pantalla un mayor número de resultados. Para ello haremos uso combinado de los signos de puntuación coma (,) y punto y coma (;). Como recordaremos, el primero de ellos indica un salto a la siguiente posición de tabulación y el segundo la no ejecución del retorno automático de carro al final de la sentencia **PRINT:**

```
10 FOR I=2 TO 160 STEP 2
20 PRINT I,;
30 NEXT
```

De esta forma obtenemos cuatro resultados por cada línea de impresión.

Veamos ahora otra aplicación del bucle **FOR-NEXT,** esta vez con un **STEP** de argumento fraccionario, para obtener una tabla de valores de **Y** en la función **Y=X   2** para el intervalo [−1,1], con incrementos de una centésima.



El BASIC permite incluir un bucle **FOR-NEXT** *dentro de otro, siempre que uno de ellos sea completamente interior al otro.*

```
10 FOR X=-1 TO 1 STEP .01
20 PRINT X ↑2
30 NEXT
```

Para apreciar mejor los resultados obtenidos puede pulsarse la tecla **CTRL,** con lo cual obtendremos un desplazamiento de la pantalla hacia arriba (**SCROLL**) a una velocidad veinte veces inferior.

Quizá sería lo lógico que, al finalizar la ejecución de un bucle **FOR-NEXT,** la variable contadora adoptara el valor superior del intervalo. Sin embargo esto no es así, puesto que la variable queda realmente con el valor máximo más un intervalo, es decir, con valor 21 en un bucle hasta 20 con paso 1, por ejemplo.

Esto se debe a la misma estructura **FOR-NEXT,** puesto que al alcanzarse la sentencia **NEXT** primero se incrementa la variable en el valor del paso (**STEP**) y después se compara si el nuevo valor rebasa el límite superior establecido para el bucle.



Los incrementos en los bucles FOR-NEXT (STEP) no sólo pueden ser enteros sino también decimales

ANIDAMIENTO DE BUCLES

En BASIC es perfectamente permisible el incluir un bucle **FOR-NEXT** dentro de otro, incluso más, siempre que respetemos una norma fundamental: cada uno de ellos debe ser completamente interior al otro. La siguiente estructura **FOR-NEXT** es correcta:

```
FOR I=A TO B
FOR J=C TO D
FOR K=E TO F
...
NEXT K
NEXT J
NEXT I
```

Aunque podemos escribirla también como:

```
FOR I=A TO B
FOR J=C TO D
FOR K=E TO F
NEXT K,J,I
```

En cualquier caso, podemos ver que cada una de las estructuras se encuentra completamente interior a la anterior (de rango superior). Por ello, no sería correcto por ejemplo:

```
FOR I=A TO B
FOR J=C TO D
FOR K=E TO F
...
NEXT J
NEXT K
NEXT I
```

Puesto que en este último caso los bucles no se encuentran correctamente anidados, por no ser cada uno completamente interior al anterior en secuencia.

El siguiente programa nos muestra como podemos simular el funcionamiento de un cronómetro de 24 horas apoyándonos en el anidamiento de bucles **FOR-NEXT.** De momento, con nuestro nivel de conocimientos, debemos contentarnos con una representación como esta, que ya iremos mejorando:

```
10 FOR H=0 TO 23
20 FOR M=0 TO 59
30 FOR S=0 TO 59
40 FOR C=0 TO 99
50 PRINT "   " H,M,S,C
60 NEXT C
70 NEXT S
80 NEXT M
90 NEXT H
```

Las variables **H, M, S** y **C** representan, respectivamente, las horas, minutos, segundos y centésimas, y el mensaje **CLR** que aparece en la línea 50 debe ser sustituido por el correspondiente carácter de control para el borrado de la pantalla (**CLR**), que se obtiene pulsando simultáneamente las teclas **SHIFT** y **CLR/HOME,** y se representa como un corazón en video inverso.

Es interesante que nos demos cuenta de la forma en que las variables se van incrementando, siempre desde «dentro» hacia «afuera» de los bucles, volviendo a comenzar desde cero al incrementar en uno el valor de la variable inmediatamente superior en rango. Cuando nos cansemos podemos pulsar **RUN/STOP** para detener la ejecución del programa.



Cuando nos cansemos de ver funcionar nuestro «reloj» podemos pulsar **RUN/STOP.**



PITSTOP II

El gran circo de la Fórmula 1 nos ha abierto sus puertas después de largos años de trabajo, y por fin podremos competir en todas sus reuniones, que tendrán lugar en los mejores circuitos del mundo entero.

Después de poner nuestro bólido a punto con la ayuda de un potente plantel de especialistas, nos dirigimos a la parrilla de salida. La pantalla estará dividida en dos partes: en la superior aparecerá nuestro coche, en la inferior podremos ver a nuestro adversario en la carrera. Además, un mapa del circuito nos indicará el recorrido que ya hemos realizado y la proximidad de la meta y boxes.

Nuestra elección consistirá en el número de vueltas que nos vemos capaces de realizar, la categoría de nuestro oponente y el circuito en que deseamos poner a prueba nuestra pericia. Si nos creemos ya lo suficientemente buenos y experimentados, podremos optar a competir en el gran circuito, sucesión de todos los existentes en el calendario de la Fórmula reina. En cada uno de ellos, lucharemos por lograr unos puntos que nos aupen a la primera posición al término del campeonato.



Nombre:	PITSTOP II
Distribuidor:	Compulogical
Precio:	2.300 Ptas.
Soporte:	Cinta

Los semáforos nos indicarán el momento de la salida, a partir de aquí pondremos toda la carne en el asador (2 minutos de cocción aprox.), para llegar a la meta en el menor tiempo posible. Las curvas se sucederán con rapidez y el roce de las ruedas con el asfalto (suena duro, ¿no?), las irán desgastando progresivamente, por ello es prudente entrar en boxes antes de suicidarnos en la pista.

Al llegar a boxes, situados en la llegada de cada vuelta, el tiempo seguirá corriendo, de tal modo que la rapidez en el cambio de neumáticos y llenado del tanque de gasolina puede propiciar la ganancia o pérdida de una carrera.

Con la práctica, iremos equilibrando nuestro gasto de fuel y la velocidad en cada vuelta; a velocidades muy altas nuestro consumo será mayor y las ruedas sufrirán una gran desgaste. En consecuencia, deberemos ser precavidos, de lo contrario esta carrera puede ser la última... ja, ja, ja, (siniestro a tope).

Por último, podremos elegir entre los circuitos de Brands Hatch, Hockenheim, Sebring, Watkins Glen, Rouen Les Essarts y Vallelunga.

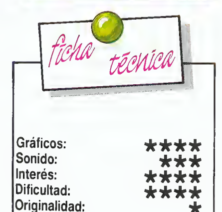
Brands Hatch tiene una longitud de 2,65 millas, con grandes rectas de alta velocidad; Hockeheim es un circuito con una gran recta, pero sus curvas son largas y peligrosas; en Sebring podremos realizar una carrera muy rápida, pero deberemos atender a sus cambios de dirección, muy cerrados; el trazado de Watkins Glen es de 3,38 millas, muy sinuoso y lento; Rouen Les Essarts es, como nos gustan a nosotros, de bonitas curvas y curioso trazado; y por último, Vallelunga, que dispone de las mayores rectas de todos los existentes, pero con las curvas más cerradas y peligrosas de todo el campeonato.

¡Atrévete a competir, sólo necesitas... un Commodore-64!.









Gráficos:	****
Sonido:	***
Interés:	****
Dificultad:	****
Originalidad:	*

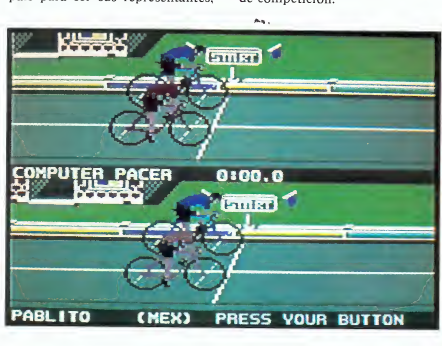
SUMMER

Sonidos de trompetas, emoción a raudales, el atleta elegido avanza su brazo y enciendo la antorcha: los Juegos Olímpicos de verano de 1984 en Los Angeles (EE.UU.)... ¡han sido inaugurados!

La multitud queda embelesada en estos precisos instantes con el grandioso espectáculo que se le brinda, y espera impaciente en sus asientos el comienzo de las pruebas del calendario olímpico.

Nosotros, elegidos por nuestro país para ser sus representantes, llegamos a estos Juegos para medirnos con los atletas más destacados del resto del mundo y lograr volver con los mayores triunfos posibles.

En este gran juego, EPYX nos proporciona la posibilidad de competir en todas las pruebas, sólo en algunas de ellas, o simplemente practicar en una o más disciplinas hasta adquirir el suficiente control necesario para optar a algún triunfo. En caso de que compitamos en alguna o todas las pruebas, podremos elegir los atletas y países que queremos tener como opositores y, a medida que vayamos señalándolos, escucharemos su himno nacional. Si nuestra elección ha sido simplemente el practicar, no gozaremos al final de la proclamación de rigor del campeón absoluto, y por supuesto, nuestras marcas no llegarán a constar como records absolutos, por considerarse realizadas fuera de competición.



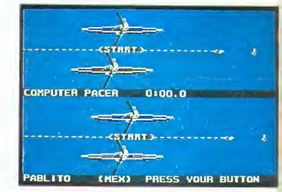
En estos modernos Juegos Olímpicos de verano habremos de luchar por conseguir medallas en las siguientes competiciones: Triple salto, Remo, Jabalina, Hípica, Salto de altura, Esgrima, Ciclismo, Kayaks.

En el Triple salto, deberemos correr hasta llegar a la línea que señala el punto de salto y realizar este lo más largo posible, para ir a caer con la pierna contraria y elevarnos de nuevo; al tocar el suelo otra vez, sólo nos restará tomar el último impulso para ir a parar al foso de arena. La longitud de nuestro intento (gozaremos de tres), se medirá desde la línea de salto hasta el punto de caída. Si el intento es lo suficientemente bueno, (a partir de 16 metros), este será repetido en pantallas gigantes de video. El salto será nulo si lo realizamos por detrás del punto señalado a tal efecto.

Nuestra siguiente prueba será el Remo, donde utilizaremos toda







GAMES II



nuestra sincronización de movimientos y fuerza. Al ver la señal de «GO», empezaremos a remar, con un ritmo continuado y similar durante toda la duración de la carrera. Intentar remar más rápido de lo posible nos hará perder segundos y posiciones, ya que las





palas de nuestros remos realizarán un menor recorrido bajo el agua, impulsándonos un menor trayecto.

Sin descansar, llega nuestro siguiente escollo, la Jabalina, en la cual procuraremos alcanzar una elevada velocidad hasta el punto donde la lanzaremos con un ángulo medio. Si este es muy abierto, lograremos que la jabalina se eleve muy alto, pero también que caiga en picado a gran velocidad; y si es cerrado, su elevación será ínfima y nuestra marca muy corta para reprimir en el público unas risas estertoreas.

¿Sabeis montar a caballo?, pues si no es así, será mejor que aprendáis pronto, puesto que aquí llega la Hípica (Equestrian), donde a lomos de un brioso corcel, deberemos saltar los obstáculos que se interpongan en nuestro camino. Cuidado con las caídas y rehuses, puesto que nuestro animal como todo, ¡ejem!... humano, tiene sus fallos.

Y a continuación el salto de altura, en el cual habrá que rebasar el fatídico listón, que caerá al más mínimo roce de nuestro cuerpo. La coordinación será en esta prueba de vital importancia, igual que en la siguiente, la Esgrima. Esquivando los golpes que nos dirija nuestro oponente con su estoque, y logrando varios «Touchés» en su pecho o cabeza (por supuesto, pertinentemente protegidos), ganaremos la lucha.

En la penúltima prueba, tendremos la ocasión de emular a Hinault o cualquier as del ciclismo, pues competiremos en un sprint que nos llevará a la meta, situada a 500 metros. Y nada más bajarnos de la bicicleta, deberemos encaminarnos hacia el río donde realizaremos la bajada en Kayak, habiendo de entrar por puertas situadas en el recorrido, y esquivar piedras e islotes, mientras luchamos con la fuerza del agua, ¡cantidad de salvaje!, ¿no?.

El infierno ya ha pasado, sólo nos resta la proclamación del mejor atleta y, como no, la triste clausura de estos Juegos Olímpicos de verano... alea jacta est.





RATON

No asustarse ni subirse a una silla, el ratón que hoy cruza las páginas de nuestra revista no es sólo inofensivo, sino además muy útil.

MIKEY MOUSE ELECTRONICO. EL PRIMER GANADOR DE NUESTRO CONCURSO DE PROGRAMACION

Felicidades a José Luis de Prat, nuestro lector barcelonés que se ha alzado con el primer premio de nuestra primera edición del concurso de programación.

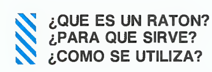
Su creación ganadora ha sido un programa de extraordinaria utilidad y corta longitud, porque ya se sabe: lo bueno, si breve, dos veces bueno.

Como podemos comprobar, nuestra revista no está sólo orientada a los juegos, y utilidades o programas de gestión tienen también un puesto muy importante en nuestras páginas, y como iremos viendo, no sólo en la sección de TECLA A TECLA o el concurso, sino también en el resto de las páginas de nuestra publicación.

El RATON es sin duda, un muy digno ganador de nuestro concurso, puesto que no sólo su breve-



dad y utilidad quedan bien patentes, sino que además ha sido desarrollado de una manera extraordinaria por su autor, obteniéndose una gran sencillez de manejo.



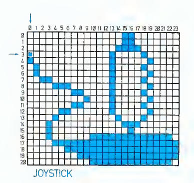
¿QUE ES UN RATON? ¿PARA QUE SIRVE? ¿COMO SE UTILIZA?

Ante la enorme curiosidad del redactor del título de este epígrafe, que pregunta más que un catecismo, nos pondremos manos a la obra aclarando cada una de sus interrogantes.

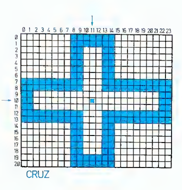
El ratón, aparte por supuesto de un roedor, es un simpático periférico que facilita enormemente la toma de datos para determinados programas. Consiste en una pequeña caja, dotada generalmente de dos botones, que manejada con la mano puede deslizarse sobre una superficie que simula la pantalla del ordenador.

Si los movimientos de un cursor de cualquier tipo en la pantalla, se hacen coincidir con las evoluciones del ratón, podremos conseguir un efecto de toma de datos bastante más fácilmente comprensible y controlable, sobre todo por el operador inexperto.

Supongamos por ejemplo que en un programa de geografía se nos presenta un mapa de Europa en la pantalla, solicitándosenos que indiquemos el punto del mapa del cual deseamos a que país pertenece. Será realmente complicado señalarlo, si no empleamos algún tipo de cursor para ello (una cruz, un punto, un cuadrado, etc...), y no cabe duda que el movimiento de este será más cómodo gracias a un ratón que mediante determinadas teclas.



JOYSTICK



CRUZ

UN RATON CON ASPECTO DE JOYSTICK, PERO QUE REALMENTE ES MUCHO MAS QUE ESO

Para simular el periférico ratón, emplearemos en nuestro caso un joystick, con el cual se conseguirá también un efecto muy aceptable; ahora bien, muchos de nosotros somos capaces de preparar una pequeña rutina que mueva un cursor por la pantalla según los datos de un joystick, así pues, ¿qué tiene de especial este programa?

Sin duda su punto más favorable es como nos facilita el trabajo a los programadores, puesto que se trata de una rutina en código máquina, controlada durante la interrupción, de forma que una vez ejecutada, podemos despreocuparnos totalmente del movimiento del joystick, dado que este será controlado de forma automática.

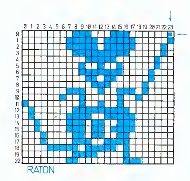
Pero aún hay más, las variables RX, RY y RF, serán también actualizadas en tiempo de interrupción, de forma que en cualquier momento podremos saber la posición del cursor dirigido por el joystick, así como la situación del pulsador de disparo.

Como ya habréis supuesto los más perspicaces, el significado de las variables es el siguiente: RX (RATON X) posición del cursor en horizontal, RY (RATON Y) posición del cursor vertical y, RF (RATON FIRE) estado del botón de disparo. La primera podrá adoptar valores entre 0 y 319, la segunda, de 0 a 199, y por último RF, 1 ó 0, según el botón esté pulsado o no, respectivamente.

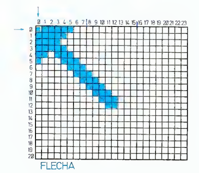
Lógicamente, estas variables pueden ser empleadas a nuestra conveniencia como cualquier otra variable BASIC. Por ejemplo, podemos emplear algoritmos del tipo **CO=INT(RX/8)** para obtener la columna en que se sitúa el cursor (de 0 a 39) o **FI=INT(RY/8)** para averiguar la fila (de 0 a 24).

SISTEMA DE ADOPCION DEL RATON HUERFANITO DE USUARIO RESIDENTE EN LOS LISTADOS DE ESTA SECCION

Para facilitar la introducción de la rutina proporcionamos un listado BASIC que contiene el código



RATON



FLECHA

máquina almacenado en sus sentencias **DATA**. Una vez ejecutado éste, podremos grabar la rutina generada o bien, simplemente efectuar **NEW** para destruir el BASIC y poder introducir nuestro propio programa que utilice el ratón.

Para activar la interrupción y que el sprite cursor comience a ser dirigido por un joystick conectado al port 2, debemos ejecutar **SYS 49152** y si deseamos desconectarlo simplemente tenemos que utilizar **SYS 49428** o pulsar **RUN/STOP + RESTORE.** Hemos de saber también que el ratón se desconecta automáticamente al ejecutarse los comandos **CLR, NEW** o **RUN,** aunque puede ser reactivado mediante el consabido **SYS 49152.**

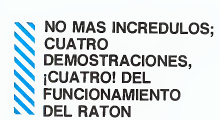
Una de las características estéticamente más sobresalientes es la forma escogida para el ratón-cursor. El programa original contiene una flecha que indica la diagonal superior izquierda, aunque nos hemos permitido incluir tres «mini-listados», añadidos al soporte BASIC original, nos podrán proporcionar diferentes cursores: en forma de cruz, a modo de simpático ratoncillo con una ceremoniosa pajarita e indicativo puntero, y como no, de joystick propiamente dicho.

Como podemos apreciar, el sistema seguido para alterar la forma del ratón es bien simple. Sólo hemos de incluir en la línea 340 un bucle del tipo **FOR I=1 TO 63: READ A: POKE 49346+I,A: NEXT** y a partir de 420 una serie de **DATAS** que contengan 63 bytes con la nueva forma del cursor.

Como las dimensiones de un sprite son considerables, para que la posición en que se encuentra el ratón sea suficientemente clara, hemos de escoger un punto del mismo que sirva para determinarla. Así por ejemplo, en el caso del cursor flecha se ha elegido el 0,0 correspondiente a la punta de misma (esquina superior izquierda), para el ratón el extremo del puntero, para la cruz su centro y para el joystick el final del cable.

Como es lógico, también este punto de referencia debe poder ser alterado; en este caso se emplean dos simples **POKES** añadidos en la línea 345: **POKE 49410,X: POKE 49411, Y,** donde X e Y deben ser sustituidos, respectivamente, por los valores de coordenadas horizontal y vertical del punto de referencia.

Finalmente hay que decir, que el cursor del ratón se ubica en el sprite 7, no siendo aconsejable cambiar sus atributos (forma, tamaño...) mediante los **POKES** habituales, pues RX y RY no facilitarían resultados correctos. Tan solo el color puede ser alterado normalmente mediante **POKE** en 53294.



NO MAS INCREDULOS; CUATRO DEMOSTRACIONES, ¡CUATRO! DEL FUNCIONAMIENTO DEL RATON

Para los incrédulos o los ansiosos por probar de inmediato las excelencias de este ratón, una vez seleccionada la forma deseada, y cargado el correspondiente código máquina, mediante la ejecución del soporte BASIC, pueden proceder a la introducción del último de los listados, que es una demostración palpable, a través de cuatro ejemplos, de las excelencias de esta rutina.

La DEMO-1 simplemente nos muestra los valores de las variables RX, RY y RF, pasando a la siguiente demostración con sólo pulsar el botón de fuego con el cursor situado sobre el cuadrado representado en la pantalla.

La segunda demostración emplea el ratón en una toma de datos de tipo menú, en él podemos efectuar la elección mediante el posicionamiento del cursor y la pulsación de fuego. La función definida en la línea 105, nos permite conocer en cualquier momento el código de pantalla del carácter sobre el cual se sitúa el cursor.

La tercera demostración va algo más lejos. En la esquina superior izquierda aparecerá el carácter que tengamos bajo el cursor gráfico; a su vez, en la parte baja se escribirá el código de pantalla de dicho carácter. Es reseñable la extraordinaria facilidad con que el programa trata el fin de DEMO-3: un simple **IF** en la línea 250 se encarga de todo.

Por último, la DEMO-4 está destinada a aquellos que encuentran pequeños los caracteres normales de pantalla y solicitan una mayor legibilidad de los mismos.

Tengamos en cuenta asimismo que la línea 30 del programa de demostración contiene la carga del código máquina desde disco (comprobando previamente mediante un **PEEK** si no se encuentra ya residente en la memoria); para alterar esta circunstancia y que la rutina se cargue desde cinta, hemos de sustituir la instrucción **30... LOAD "RATON",8,1** por **30... LOAD"RATON",1,1.**

CURSOR CRUZ

```
340 FORI=1TO63:READA:POKE49346+I,A:NEXT              -197-
345 POKE49410,11:49411,10                            -069-
420 DATA0,254,0,0,254,0,0,198,0,0,198,0,0,198,0,0
                                                     -144-
425 DATA198,0,0,198,0,255,199,255,255,199,255,192,0,3
,192,0                                               -190-
430 DATA3,192,0,3,255,199,255,255,199,255,0,198,0,0,1
98,0                                                 -081-
435 DATA0,198,0,0,198,0,0,198,0,0,254,0,0,254,0,0
                                                     -150-
```

CURSOR JOYSTICK

```
340 FORI=1TO63:READA:POKE49346+I,A:NEXT              -197-
345 POKE49410,0:POKE49411,3                          -020-
420 DATA0,1,128,0,1,128,0,3,192,128,4,32,128,4,16,64
                                                     -047-
425 DATA4,32,64,4,16,48,4,32,14,4,16,1,132,32,0,68
                                                     -214-
430 DATA16,1,132,32,6,4,16,8,2,32,16,1,192,16,0,128
                                                     -254-
435 DATA16,63,254,15,255,255,1,255,255,0,127,255,0,63
,254,0                                               -166-
```

CURSOR RATON

```
340 FORI=1TO63:READA:POKE49346+I,A:NEXT              -197-
345 POKE49410,23:POKE49411,0                         -070-
420 DATA0,227,129,1,247,193,1,247,194,0,255,130,0,62,
4,0                                                   -254-
425 DATA73,4,0,127,8,0,54,8,0,28,16,1,201,208,2,65
                                                      -214-
430 DATA32,128,73,0,64,255,128,65,231,192,33,217,192,
33,219,192                                            -119-
435 DATA16,217,128,16,231,128,14,190,128,1,240,128,3,
128,224,0                                             -048-
```

SOPORTE BASIC

```
1 REM***********************************              -201-
2 REM*****       RATON 64        *********            -016-
3 REM*****  (C) J.L. DE PRAT     *********            -169-
4 REM*****    ENERO - 1986       *********            -066-
5 REM***********************************              -205-
7 REM                                                 -059-
8 REM **      CARGADOR       BASIC      **            -169-
9 REM                                                 -061-
10 PRINT"{CLR}{ABJ}{YEL}[13 ESP]RATON-64[10 ESP]{BLK}
(C) JOLI{YEL}":PRINT"{ABJ}CARGANDO DATAS"             -135-
15 DATA162,0,134,13,134,14,189,8,193,133,69,232,189,8
,193,133,70,134,2,32                                  -094-
20 DATA231,176,166,2,157,13,193,152,157,14,193,232,22
4,6,208,226,173,18                                    -253-
30 DATA193;133,251,173,19,193,133,252,172,15,193,174,
14,193,208,1,136,202                                  -091-
40 DATA142,45,193,140,46,193,162,0,189,195,192,157,19
2,2,232,224,63,208                                    -006-
50 DATA245,169,11,141,255,7,169,1,141,46,208,173,29,2
08,41,127,141,29,208                                  -098-
60 DATA173,23,208,41,127,141,23,208,173,28,208,41,127
,141,28,208,173,16                                    -244-
70 DATA208,41,127,141,16,208,169,159,141,14,208,169,1
04,141,15,208,173,21                                  -090-
80 DATA208,9,128,141,21,208,169,0,141,27,208,169,24,5
6,237,2,193,141,4,193                                 -147-
90 DATA169,87,56,237,2,193,141,5,193,169,50,56,237,3,
193,141,6,193,169,249                                 -184-
100 DATA56,237,3,193,141,7,193,32,40,194,32,248,193,1
20,162,44,160,193,142                                 -138-
110 DATA20,3,140,21,3,88,96,252,0,0,248,0,0,240,0,0,2
48,0,0,220,0,0,142,0                                  -005-
120 DATA0,7,0,0,3,128,0,1,192,0,0,224,0,0,112,0,0,56,
0,0,24,0,0,0,0,0,0,0                                  -208-
130 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,255,2
55,255,255,82,88,82                                   -186-
140 DATA89,82,70,5,8,5,8,5,8,56,120,162,49,160,234,14
2,20,3,140,21,3,144                                   -024-
150 DATA1,88,173,21,208,41,127,141,21,208,96,173,255,
255,201,88,208,12,165                                 -145-
160 DATA46,197,48,208,13,165,45,197,47,208,7,24,32,22
,193,76,49,234,173,0                                  -118-
170 DATA220,9,224,73,255,141,19,3,173,19,3,41,16,240,
9,169,129,160,0,145                                   -036-
180 DATA251,24,144,6,169,0,160,0,145,251,173,19,3,41,
239,208,3,76,49,234                                   -040-
190 DATA173,19,3,41,2,240,17,172,15,208,204,7,193,240
,30,238,15,208,32,248                                 -130-
200 DATA193,76,157,193,173,19,3,41,1,240,14,172,15,20
8,204,6,193,240,6,206                                 -132-
210 DATA15,208,32,248,193,173,19,3,41,8,240,34,173,16
,208,41,128,240,8,174                                 -137-
220 DATA14,208,236,5,193,240,66,238,14,208,208,8,173,
16,208,9,128,141,16                                   -047-
230 DATA208,32,40,194,76,49,234,173,19,3,41,4,240,40,
173,16,208,41,128,240                                 -132-
240 DATA19,174,14,208,208,8,173,16,208,41,127,141,16,
208,206,14,208,24,144                                 -135-
250 DATA11,174,14,208,236,4,193,240,6,206,14,208,32,4
0,194,76,49,234,173                                   -042-
260 DATA16,193,133,253,173,17,193,133,254,173,15,208,
56,237,6,193,24,160                                   -055-
270 DATA1,162,7,41,255,240,9,48,4,10,202,208,246,41,1
27,56,145,253,144,4                                   -030-
280 DATA138,24,105,129,160,0,145,253,96,173,14,193,13
3,253,173,15,193,133                                  -097-
290 DATA254,160,2,169,0,145,253,173,14,208,56,237,4,1
93,168,173,16,208,41                                  -103-
300 DATA128,240,27,152,48,24,74,133,2,144,6,169,128,1
60,2,145,253,160,1,165                                -180-
310 DATA2,145,253,160,0,169,137,145,253,96,152,76,9,1
94                                                    -241-
320 FORX=0TO613:READC:POKE49152+X,C:S=S+C:NEXT
                                                      -155-
330 IF S<>66933 THEN PRINT"ERROR EN DATAS!!":END
                                                      -154-
350 PRINT"QUIERES GRABAR LA RUTINA DE CODIGO MAQUINA?
 (S/N)"                                               -064-
360 INPUT A$:IFA$<>"S"ANDA$<>"N"THEN350               -024-
370 IF A$="N"THEN END                                 -195-
380 PRINT"{ABJ}PERIFERICO: 1=CINTA, 8=DISCO"
                                                      -239-
390 INPUT P:IF P<>1ANDP<>8THEN380                     -063-
400 POKE 251,P                                        -023-
410 POKE43,0:POKE44,192:POKE45,103:POKE46,194
                                                      -111-
415 SAVE"RATON",PEEK(251),1:POKE43,1:POKE44,8:POKE45,
32:POKE46,19                                          -118-
```

DEMOSTRACION

```
10 REM ******  DEMO RATON 64 **********               -216-
20 REM                                                -102-
30 IF PEEK(49152)<>162 THEN LOAD"RATON",8,1           -216-
35 SYS49152:REM ACTIVA RATON                          -038-
40 PRINT"{CLR}{YEL}[16 ESP]DEMO1"                     -220-
45 PRINT"MUEVE EL RATON POR LA PANTALLA PARA VER COMO
 CAMBIA EL VALOR DE";                                 -241-
47 PRINT" LAS VARIABLES."                             -067-
50 PRINT"{HOM}{4 ABJ} RX="RX;"{IZQ} ","RY=";RY;"{IZQ}
 ","RF=";RF                                           -044-
55 PRINT"{2 ABJ}CONTINUAR {RON} {ROF}"                -020-
60 IF RF=0 THEN 50 :REM ESPERA HASTA QUE SE PULSE FUE
GO                                                    -191-
70 COL=INT(RX/8):FILA=INT(RY/8)                       -208-
80 IF CO<>10ORFI<>7 THEN50                            -057-
90 PRINT"{CLR}[16 ESP]DEMO2"                          -068-
100 PRINT"{ABJ}ESCOGE UNA OPCION PONIENDO LA FLECHA E
N-";                                                  -060-
103 PRINT"CIMA DEL NUMERO Y PULSA DISPARO"            -218-
105 DEFFNC(X)=PEEK(INT(RX/8)+INT(RY/8)*40+1024)
                                                      -254-
110 PRINT:PRINT"[6 ESP]{RON}1{ROF} - OPERACIONES VARI
AS{ABJ}"                                              -123-
120 PRINT"[6 ESP]{RON}2{ROF} - SEGUNDA SELECCION{ABJ}
"                                                     -084-
130 PRINT"[6 ESP]{RON}3{ROF} - TERCERA OPCION{ABJ}"
                                                      -136-
140 PRINT"[6 ESP]{RON}4{ROF} - CONTINUAR CON DEMO{2 A
BJ}"                                                  -165-
150 IF RF=0 THEN 150 :REM ESPERA HASTA QUE SE PULSE F
UEGO                                                  -032-
160 C= FN C(0): REM C=CARACTER BAJO EL CURSOR
                                                      -221-
170 C=C-128:IF C<49 OR C>52 THEN 150:REM 49=COGIGO1,5
2=CODIGO4                                             -112-
180 PRINT"{ARB}HAS ESCOGIDO LA OPCION";C-48           -016-
190 IF C<>52 THEN 150                                 -146-
200 PRINT"{CLR}[16 ESP]DEMO3"                         -110-
210 PRINT:FORI=32TO127:PRINTCHR$(I);" ";:NEXT:PRINT:P
RINT                                                  -204-
220 PRINT"{2 ABJ}MUEVE EL CURSOR POR LA PANTALLA PARA
 VER EL CODIGO DEL CARACTER{2 ABJ}"                   -086-
230 PRINT"CONTINUAR[2 ESP]{RON} {ROF}{2 ABJ}"
                                                      -095-
235 PRINT"{ARB}CODIGO ";FN C(0);"{IZQ} "              -224-
237 POKE1024,FNC(0)                                   -054-
240 IF RF=0 THEN 235                                  -115-
250 IF FNC(0)<>160THEN 235                            -184-
260 PRINT"{CLR}[16 ESP]DEMO4"                         -117-
270 PRINT"{ABJ}OTRA FORMA DISTINTA DE MENUS:{ABJ}"
                                                      -130-
280 A$="{RON}[3 ESP]{ABJ}{3 IZQ}[3 ESP]{ABJ}{3 IZQ}[3
 ESP]{ARB}{ROF}"                                      -197-
290 PRINT:PRINTTAB(5)A$;" - PRIMERA OPCION{3 ABJ}"
                                                      -232-
300 PRINTTAB(5)A$;" - ESTA ES LA SEGUNDA{3 ABJ}"
                                                      -218-
310 PRINTTAB(5)A$;" - Y LA ULTIMA{3 ABJ}"             -020-
320 PRINTTAB(5)A$;" - FIN DEMO"                       -018-
330 IF RF=0 THEN 330                                  -111-
340 IF FNC(0)<>160THENPRINT"{HOM}NADA[5 ESP]":GOTO330
:REM SI CAR.NO ES ESP.INV.VOLVER                      -033-
345 COL=INT(RX/8)::IF CO<5ORCO>7THEN400               -181-
350 FILA=INT(RY/8)                                    -095-
360 IF FI>4ANDFI<8THEN O=1                            -075-
370 IF FI>9ANDFI<13THEN O=2                           -126-
380 IF FI>14ANDFI<18THEN O=3                          -177-
390 IF FI>19ANDFI<23THEN PRINT"{CLR}FIN DEMO":SYS 494
28:END                                                -239-
400 PRINT"{HOM}{WHT}OPCION ";O:GOTO 330               -056-
```

¡FUERA E

C-Byte tiene el honor de presentar, en exclusiva nacional para los lectores de TU MICRO COMMODORE, el revolucionario sistema de introducción de programas FUERA ERRORES. Este nos permitirá introducir, sin temor alguno al esfuerzo inútil, cualquier listado por largo y complicado que parezca.

Para adoptar los listados publicados bajo este sistema, deberemos seguir las siguientes normas:

1) Es fundamental transcribir EXACTAMENTE el listado reproducido, incluyendo todos sus espacios, aunque se trate de separaciones entre número de instrucción y línea de instrucción.

2) Todas las líneas finalizarán con un número de tres dígitos, encerrado entre guiones, que NO deberá ser introducido, puesto que no forma parte del programa, sino que tiene la finalidad de hacer funcionar el sistema FUERA ERRORES, según veremos más adelante. Para evitar equivocaciones, dicha cifra entre guiones se sitúa en el margen derecho del final de la línea BASIC a la cual corresponde, a una distancia prudencial del mismo.

3) Para facilitar la introducción de símbolos difícilmente interpretables, se procede a la siguiente representación en los listados.

— Las letras aparecidas entre menor y mayor deberán ser introducidas con pulsación simultánea de la tecla COMMODORE y la letra representada. Ej.: <M> =COMMODORE M.

— Las letras aparecidas entre barras verticales deberán ser introducidas como pulsación simultánea de la tecla SHIFT y la letra representada. Ej.: ¦ K ¦ = SHIFT K.

— Entre corchetes simples se representarán los símbolos que se obtienen por pulsación directa de la tecla, aunque lógicamente, este caso sólo se dará para indicar las sucesiones de más de una letra. Así por ejemplo, la introducción de 5 asteriscos se representaría por [5*].

— Para la repetición de símbolos obtenidos mediante las teclas COMMODORE o SHIFT, se seguirá una combinación de las tres normas anteriormente citadas. Así por ejemplo, la introducción de 10 símbolos COMMODORE H, se representaría por [<10 H>].

— Para evitar confusiones, cuando se utilice el sistema de representación de sucesiones de carácter, y éste sea un espacio, se utilizará la abreviatura ESP. [15 ESP] = 15 espacios.

— Los carácteres de control, tales como desplazamientos del cursor, colores, estados de reversa y funciones, se simbolizarán por una abreviatura de tres letras (dos más un espacio en el caso de las funciones) encerrada entre llaves, tal como se señala en la tabla adjunta.

Para introducir cualquier listado por el sistema FUERA ERRORES, deberemos entrar previamente y ejecutar el listado BASIC que aparece en la página siguiente por lo cual es recomendable conservar una copia grabada del mismo, para sucesivas ocasiones. Una vez introducido este listado, ya sea por el teclado, o a través de cinta o disco, debemos ejecutarlo con **RUN.** Instantes más tarde aparecerá en la pantalla el mensaje **FUERA ERRORES!** y el cursor libre para la entrada de programas, con el tradicional **READY.** por encima, lo cual indicará la activación del sistema de depuración de errores.

En virtud al **NEW** que finaliza la línea 20 del programa ¡FUERA ERRORES!, este habrá desaparecido de la memoria, y seremos libres para introducir cualquiera de los programas listados en la sección **TECLA A TECLA** de cualquier número de nuestra revista, o incluso aquellos que apareciendo en otras secciones se acojan a este sistema. Así pues, si el programa ¡FUERA ERRORES! ha desaparecido de la memoria, ¿qué hemos conseguido ejecutándolo? Bien, la respuesta se llama informáticamente **INTERRUPCION:** se trata de una técnica de programación en código máquina que permite que el ordenador efectúe prácticamente dos trabajos a un tiempo, o más correctamente, que ejecute determinada tarea de forma automática, sin necesidad de que le prestemos una atención constante, de forma similar al proceso de respiración en un humano.

Efectivamente, aunque el soporte BASIC ha desaparecido de la memoria, antes de «marcharse» ha dejado funcionando en modo interrupción la pequeña rutina en código máquina que se hallaba en sus **DATAS.** Para comprobarlo pulsa **RETURN**; observarás algo muy extraño: tu ordenador no se comporta normalmente, no sólo desciende una línea el cursor, sino que además hace aparecer un número en la esquina superior izquierda de la pantalla. A continuación veremos como emplearlo.

Cada vez que pulsemos **RETURN**, aparecerá un número en la mencionada zona de la pantalla, y este corresponderá con la instrucción que hayamos introducido. Esto forma parte del sistema de FUERA ERRORES. Cuando introduzcamos cualquier línea de un listado de este tipo, deberemos fijarnos en el número que aparece al pulsar el **RETURN** de fin de línea; si este coincide con el que aprece en el listado al final de la línea, ésta habrá sido introducida correctamente, en caso contrario existe algún error de teclado que debemos modificar. Para modificar una instrucción errónea, no tenemos ni tan siquiera que volver a teclearla si no queremos, bastará sencillamente con modificar el carácter o caracteres erróneos como siempre hacemos, hasta que nos coincida el número de verificación que se presentará al pulsar **RETURN.**

Así pues, el sistema FUERA ERRORES se compone de dos partes: una codificación especial de los listados que facilitan su introducción, evitando los errores al confundir los caracteres gráficos, de control, etc... y un sistema de verificación de líneas que nos advierte en el preciso instante de introducir una de estas, que está mal tecleada.

Ahora bien, el empleo de estos dos sistemas no quiere decir que nos encontremos ante un BASIC diferente al de COMMODORE 64. Este no ha cambiado, hace exactamente las mismas cosas de siempre; simplemente hemos cambiado la forma de hacer los listados. En cuanto al misterioso número que aparece en la esquina de la pantalla no es más que

RRORES!

una simple suma de comprobación, lo que se conoce técnicamente como un CHECKSUM. La rutina en código máquina de interrupción suma los valores de los caracteres que entramos en cada línea que introducimos, aplicándoles un módulo 256, es decir, volviendo a cero cada vez que su suma parcial supera el 255, de este modo se crea un número entre 0 y 255 dependiente directamente de los caracteres introducidos. Así pues, es prácticamente imposible que una sucesión de errores den por casualidad ese número, mientras que siempre que la línea esté correctamente introducida obtendremos el mismo código que nosotros hallamos por igual sistema y añadimos al final de cada línea al realizar el listado del programa.

Por tanto, el misterioso número no es tampoco una modificación del BASIC de COMMODORE, sino simplemente un pequeño truco para la comprobación de que las líneas han sido bien introducidas. Propiamente no nos evita cometer errores de tecleado, sino que simplemente nos advierte inmediatamente en que línea los hemos introducido.

Para desactivar el sistema sólo deberemos pulsar RUN/STOP RESTORE, y si por cualquier motivo nos interesara reactivarlo, podríamos ejecutar SYS 822, siempre y cuando se encuentre el código máquina en la memoria, lógicamente.

¡ADVERTENCIA! Puesto que el código máquina se encuentra ubicado en el buffer del casete, es imprescindible desactivarlo (RUN/STOP RESTORE) antes de realizar cualquier operación con dicho periférico.

UTILIZACION DEL CASETE CON FUERA ERRORES

Si queremos introducir parte de un programa, para continuar posteriormente el trabajo emprendido, sin perder por supuesto la enorme ventaja del FUERA ERRORES deberemos proceder de la siguiente forma:

1. Desactivar el sistema FUERA ERRORES mediante la pulsación de RUN/STOP + RESTORE.

2. Trasladar el código máquina desde su ubicación en el buffer del casete hasta otro punto desocupado en la memoria RAM (por ejemplo 49152). Para ello sólo hemos de saber que ocupa 114 bytes desde 822 (inclusive). Una sencilla rutina que efectúe el trabajo de reubicación puede ser: FORI=0TO113:POKE49152+I,PEEK(822+I):NEXT.

3. Realizar la operación correspondiente con el casete; ya sea grabar una copia de seguridad de lo introducido en la memoria hasta el momento, ya sea cargar del casete una parte ya grabada del programa.

4. Restablecer FUERA ERRORES a su punto original. Siguiendo el caso de la línea de ejemplo anterior: FORI=0TO113:POKE822+I,PEEK(49152+I):NEXT.

5. Reactivar el sistema FUERA ERRORES mediante SYS 822 (esta vez no aparecerá mensaje alguno, aunque al pulsar **RETURN** comprobaremos que aparecen las cifras de control en la esquina superior izquierda). Lógicamente, los pasos 4 y 5 no son necesarios si después de una grabación se va a apagar el ordenador, y sólo se llevarán a cabo después de las cargas, o si tras la grabación de una copia de seguridad del programa, deseamos continuar introduciéndolo acto seguido.

TABLA DE INTERPRETACION DE CODIGOS DE CONTROL

ABR	SIFNIFICADO	OBTENCION	ABR	SIFNIFICADO	OBTENCION
HOM	HOME	CLR/HOME	F8	FUNCION 8	SHIFT F7
CLR	CLEAR + HOME	SHIFT CLR/HOME	BLK	BLACK (NEGRO)	CTRL 1
ABJ	CURSOR ABAJO	CRSR VERTICAL	WHT	WHITE (BLANCO)	CTRL 2
ARB	CURSOR ARRIBA	SHIFT CRSR VERTICAL	RED	RED (ROJO)	CTRL 3
			CYN	CYAN (CIAN)	CTRL 4
DCH	CURSOR DERECHA	CRSR HORIZONTAL	PUR	PURPPLE (PURPURA)	CTRL 5
IZQ	CURSOR IZQUIERDA	SHIFT CSRS HORIZONTAL	GRN	GREEN (VERDE)	CTRL 6
			BLU	BLUE (AZUL)	CTRL 7
RON	REVERSE ON	CTRL 9	YEL	YELLOW (AMARILLO)	CTRL 8
ROF	REVERSE OFF	CTRL 0	NRJ	NARANJA	COMMODORE 1
F1	FUNCION 1	F1	MRN	MARRON	COMMODORE 2
F2	FUNCION 2	SHIFT F1	RCL	ROJO CLARO	COMMODORE 3
F3	FUNCION 3	F3	GR1	GRIS 1	COMMODORE 4
F4	FUNCION 4	SHIFT F3	GR2	GRIS 2	COMMODORE 5
F5	FUNCION 5	F5	VCL	VERDE CLARO	COMMODORE 6
F6	FUNCION 6	SHIFT F5	ACL	AZUL CLARO	COMMODORE 7
F7	FUNCION 7	F7	GR3	GRIS 3	COMMODORE 8

```
10 FORI=822TO935:READA:C=C+A:POKEI,A:NEXT
20 IFC<>15254THENPRINTCHR$(147)"ATENCION!, HAY UN ERROR EN LOS DATOS":END
30 PRINTCHR$(147)TAB(213)"FUERA ERRORES!":SYS822:NEW
100 DATA 169,3,141,37,3,169,69,141,36,3,169,0,133,254,96,32,87,241,133,251
110 DATA 134,252,132,253,8,201,13,240,13,24,101,254,133,254,165,251,166,252
120 DATA 164,253,40,96,169,13,32,210,255,165,214,141,176,3,206,176,3,169,0
130 DATA 133,216,169,18,32,210,255,169,19,32,210,255,169,45,32,210,255,166
140 DATA 254,224,100,176,5,169,48,32,210,255,224,10,176,5,169,48,32,210,255
150 DATA 169,0,133,254,32,205,189,169,45,32,210,255,173,176,3,133,214,76,88,3
```

CODIGO MAQUINA

Cada vez que se menciona el código máquina, se hace con un cierto aire de misterio, como algo complicado y fuera del alcance de todo el mundo. Nada más lejos de la realidad. Bienvenidos desde hoy al mundo del código máquina.

Un lenguaje que todos, más o menos, dominamos es el BASIC. Los lenguajes de este tipo que nos permiten controlar a nuestro ordenador en un lenguaje similar al humano (al humano inglés, claro) se denominan LENGUAJES DE ALTO NIVEL. Casi todos los códigos que se emplean en ordenadores personales son de alto nivel, principalmente para simplificar su uso y el aprendizaje en ellos.

Las instrucciones de estos lenguajes son simbólicas y de gran potencia; así, para expresar una suma sólo tenemos que escribir algo como esto:

```
10 LET A=B+C
```

Sencillo, ¿verdad?. El gran problema es que el ordenador no entiende esto tal como lo hemos escrito, porque cualquier ordenador (sí, incluso nuestro Commodore) sólo entiende su propio lenguaje máquina.

Este código es un conjunto de instrucciones sencillas, que el ordenador comprende «de fábrica», y que pueden ser del tipo de leer o escribir a memoria, comparar dos números, efectuar un salto en el programa, etc. Cada operación viene representada por un número, denominado CODIGO DE OPERACION, y uno o dos más que indican otros datos sobre dicha operación, como por ejemplo, la posición de memoria con la cual debe operar a los datos que debe emplear.

Es fácil comprender que nos movemos en dos mundos diferentes, el humano, simbólico y «de letras», y el del ordenador, determinista y totalmente numérico. La máquina no puede trabajar para nosotros si no le hablamos en su propio idioma, y a nosotros nos es muy pesado utilizar un lenguaje numérico tan distinto del nuestro.

La solución está en escribir programas traductores, que conviertan unas instrucciones en lenguaje más o menos natural, a su equivalente de código máquina que el ordenador pueda comprender y ejecutar. Naturalmente, estos traductores deberán estar escritos en lenguaje máquina, pero una vez terminados en teoría no es necesario volver a emplearlo. Decimos en teoría, pues ya veremos que a veces no nos queda otro remedio.

Existen dos formas de llevar esta idea a la práctica: los llamados COMPILADORES y los INTERPRETES.

Los compiladores toman el programa escrito en un lenguaje de alto nivel y lo traducen por entero a C/M (código máquina), es decir, transforman un código «fuente» (lo que nosotros suministramos al ordenador, escrito en forma clara para el humano) en un código «objeto» (lo que el ordenador entiende, aunque sería incomprensible para nosotros). Este último es el que en realidad se utiliza, pues el código fuente sólo sirve para producir el objeto, y se pierde o se almacena por separado.

Un intérprete hace lo mismo, pero traduciendo una instrucción del programa y ejecutándola a continuación, para tomar la siguiente y repetir el proceso. Ejemplos de lenguajes que emplean intérpretes son el BASIC y el LOGO, mientras otros como el PASCAL o el FORTRAN, utilizan compiladores. Esto no quiere decir que no haya versiones de BASIC compiladas o intérpretes de PASCAL, simplemente que en el momento de su creación, sus diseñadores los concibieron dentro de un tipo u otro de lenguaje: compilados o interpretados.

¿Qué sistema es preferible? Un compilador produce programas unas 10 veces más rápidos que sus equivalentes interpretados; además, como el compilador sólo es necesario cuando se compila (se traduce) el programa, la memoria que ocupa debe residir en memoria al tiempo que el programa, consumiendo, por ejemplo, 8 K del C-64, por lo que los programas deberán ser más cortos de lo que la memoria nos permite en principio.

Sin embargo, usar un intérprete es más sencillo a la hora de poner a punto un programa, o cuando es necesario modificarlo; imaginemos lo que sería que al corregir un programa, por cada cambio hubiera que esperar a que el compilador lo tradujese todo para a continuación indicarnos que hay un

error, y tener que volver al principio. Así para todo un programa. Es principalmente por esta razón por la que la mayoría de los ordenadores personales están dotados de intérpretes. Además, un compilador suele necesitar de un dispositivo de almacenamiento, como una unidad de disco, durante el proceso de compilación, mientras que es posible operar un intérprete con un medio más sencillo, como el cassette.

SI NOS RESULTA MAS SENCILLO PROGRAMAR EN BASIC ¿POR QUE MOLESTARNOS EN EMPLEAR CODIGO MAQUINA?

Bien, ya tenemos el intérprete o compilador funcionando en nuestra máquina. Si podemos programar en un lenguaje sencillo, ¿por qué complicarnos con el código máquina? Pues porque, sencillamente, a veces no queda otro remedio. Por ejemplo, un programador no muy hábil puede ahorrar hasta un 50 % de memoria, y alcanzar velocidades de a 6 veces las de un programa equivalente compilado, y el intérprete ni lo mencionamos en cuestiones de velocidad.

Además, para acceder a los recursos más internos de la máquina, como programar periféricos, interfaces, manipular la pantalla, etc, casi siempre el único camino para hacerlo con efectividad es el C/M. Y en el mundo de los juegos, el código máquina es imprescindible si queremos conseguir buena movilidad, rapidez o simplemente, unos gráficos espectaculares.

Aunque hasta ahora hemos hablado de programar en código máquina. Realmente esto no se hace casi nunca directamente, pues sería pesado en exceso. Existen unos programas, llamados ENSAMBLADORES, que nos ayudan en esta tarea; básicamente, evitan tener que recordar la tabla de códigos del ordenador, sustituyendo cada instrucción por un conjunto de, en general, tres letrras que forman un código MNEMONICO de la instrucción. Disponemos además de una serie de ayudas, como el uso de variables, o saltos a posiciones de memoria que previamente hemos identificado con un nombre. Aunque ahora esto pueda parecer algo raro, ya veremos más adelante que representan una simplificación muy grande.

Existen otros programas de ayuda, como los MONITORES. Un monitor nos permite depurar un programa en lenguaje máquina ya escrito, pero no funciona correctamente. Podemos ejecutarlo paso a paso y ver el resultado de cada operación en la memoria o en los registros del ordenador.

Gracias a los ensambladores, los monitores y toda una serie de programas de ayuda, no es excesivamente difícil comenzar a explorar las profundidades de nuestro ordenador. Todo el mundo de posibilidades que se abre ante nosotros es el que hoy hemos comenzado a explorar. Bienvenidos a bordo.



C O N C U R S O

El travieso C-Byte tiene el honor de invitaros a la participación en nuestro primer concurso de programación. Los requisitos necesarios son bien pocos:

- Saber programar un ordenador COMMODORE.
- Ser español o extranjero y
- Tener una edad comprendida entre 5 y 105 años.

Fácil, ¿verdad?
En cuanto a los premios, la mar de atractivos:

- **1.er premio.–60.000 pesetas en material informático a escoger por el premiado, más un póster de C-Byte.**
- **2.° premio.–30.000 pesetas en material informático a escoger por el premiado, más un póster de C-Byte.**
- **3.er premio.–15.000 pesetas en material informático a escoger por el premiado, más un poster de C-Byte.**

Y en fin, si alguno de los programas destaca por su originalidad, estética o comicidad, no sería de extrañar que le cayera alguna cosilla más...

Las bases del concurso son bien sencillas:

1. Los programas remitidos al concurso deberán ser creación original del autor o autores, y completamente inéditos, pudiendo remitir tantos programas como se desee.

2. Los programas deberán ser enviados en casette o disco flexible a TU MICRO COMMODORE (Concurso de programación). Apartado de correos 61.294. 28080 MADRID.

3. Los programas podrán ser de cualquier tipo (juegos, utilidades, gestión, educativos) y habrán de estar escritos en lenguaje BASIC o código máquina.

4. Los programas deberán ser remitidos desprovistos de cualquier tipo de protección, que impida o dificulte el análisis del mismo, así como reproducción en estas páginas y su introducción como listado siguiendo el sistema FUERA ERRORES.

5. Cuando la ejecución del programa precise de la concurrencia de determinado periférico o aditamento (joysticks, tabletas gráficas, programas comerciales de ayuda), se valorará la indicación de las modificaciones pertinentes, para que el programa pueda ser disfrutado por cualquier usuario en la configuración básica.

6. Todo programa presentado al concurso deberá acompañarse de los siguientes datos:
   - Datos personales del concursante
   - Nombre del programa.
   - Modelo para el que está destinado.
   - Breve descripción del programa detallando las indicaciones necesarias para su ejecución.

7. Los programas premiados pasarán a ser propiedad de la revista TU MICRO COMMODORE, pudiendo hacer ésta libre uso de ellos, y renunciando sus autores a cualquier otra compensación distinta al premio.

8. Los programas no premiados que por su calidad se hagan merecedores de su publicación, serán adquiridos por la editorial, aplicando la tarifa vigente.

9. Los programas recibidos con posterioridad a la fecha tope de la presente edición, serán automáticamente incluidos en los destinados a la siguiente.

10. El jurado decidirá sobre todos los aspectos no contemplados en estas bases y su decisión será inapelable.

Y ahora a darse mucha prisa, el plazo para la recepción de programas termina el próximo día 15 de marzo.

¡¡SUERTE!!







La revista semanal donde el usuario de COMMODORE podrá encontrar gran variedad de programas y noticias, sus periféricos y equipos.

SUSCRIPCION POR TELEFONO

Todos los días, excepto sábados y festivos, de **9 a 15,00 y de 17 a 19,00 horas** atenderemos sus consultas en el ☎ 2505820/2505579.
¡Suscríbase por teléfono! y en pocos días tendrá en su domicilio TU MICRO COMMODORE.

SUSCRIPCION POR CORREO

Deseo suscribirme a la revista TU MICRO COMMODORE al precio de 7.400 ptas. **(incluido IVA),** por el período de un año a partir del N.º....... inclusive.

El importe **—que abonaré en su totalidad con el primer envío—** lo haré efectivo de la siguiente forma:

☐ Talón bancario adjunto a nombre de INGELEK, S. A. ☐ Giro Postal n.º ....
☐ Tarjeta VISA n.º |_|_|_|_| |_|_|_|_| |_|_|_|_| |_|_|_|_| FIRMA
☐ Tarjeta MASTER CARD n.º |_|_|_|_| |_|_|_|_| |_|_|_|_| |_|_|_|_|
Fecha de caducidad de la tarjeta .... Nombre titular ....

APELLIDOS |_____________| NOMBRE |_______|
DOMICILIO |_____________| EDAD |__|
CIUDAD |_______| PROVINCIA |_______|
C POSTAL |_____| TELEFONO |_______| PROFESION |_______|

Recorte o copie esta tarjeta y envíela dentro del sobre a: Ediciones INGELEK. Apartado de Correos 61.[illegible] 28080 MADRID

COMMODORE, COMPAÑIA AMERICANA, LIDER MUNDIAL EN NUMERO DE ORDENADORES INSTALADOS.

Su Commodore 64 tiene mucho que decirle.

Monitores.

El Commodore 64 es el resultado de la experiencia internacional de Commodore como líder indiscutible en el mercado de los microordenadores.

El Commodore 64 es el ordenador más completo y potente de su categoría,... pero todavía tiene mucho que decirle.

Por ejemplo, sus Monitores.

Monitores funcionales de diseño con pantalla de alta resolución.

Pensados y creados para trabajar con ellos sin notar molestias ni cansancio en la vista.

Monitores capaces de reproducir con nitidez la completa gama de colores del C-64.

Monitores con un cuidado sistema de sonido. Con conexiones conmutables de Vídeo Compuesto y Crominancia-Luminancia.

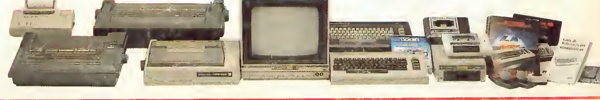
Monitores que complementan las prestaciones de su ordenador, ganando en imagen.

Amplíe las posibilidades de su C-64, descubriendo su extensa gama de periféricos.

Ahora que ya sabe que su Commodore 64 tiene todavía mucho que decirle, prepárese a conocerle mejor.

commodore 64





Microelectrónica y Control, S.A. *c/ Valencia, 49-53 08015 Barcelona - c/ Ardemans, 8 28028 Madrid.*
Unico representante de Commodore en España.